
NI-SCOPE API Overview

2022-07-06



Contents

NI High-Speed Digitizers Help.....	4
Related Documentation.....	5
Getting Started.....	9
Operating System Support.....	9
Creating an Application with NI-SCOPE.....	10
Forms of Device Synchronization.....	18
NI-SCOPE Tutorial.....	22
NI-SCOPE and InstrumentStudio.....	32
NI-SCOPE Examples.....	33
Fundamentals.....	35
Bandwidth.....	35
Flatness.....	36
Resolution.....	37
Sample Rate.....	38
Record Length.....	41
Vertical Range and Offset.....	41
Input Coupling.....	42
Impedance and Impedance Matching.....	42
Probes and Their Effects.....	44
Accuracy.....	47
Noise.....	50
Thermal Shutdown.....	51
Triggering.....	52
Clocking.....	64
Time Stamping.....	67
TDC.....	68
Sampling Methods.....	68
Frequency Domain Fundamentals.....	75
Video Signal Fundamentals.....	77
Waveform Measurements Fundamentals.....	88
Devices.....	110
State Diagrams.....	111
Features Supported by Device.....	120

Device Functionality.....	132
PXIe/PXI/PCI-5105.....	137
PXIe/PXI/PCI-5114.....	170
PXIe/PXI/PCI-5122.....	210
PXI/PCI-5124.....	256
USB-5132/5133.....	292
PXI/PCI-5142.....	311
PXI/PCI-5152.....	358
PXI/PCI-5153/5154.....	386
PXIe-5160/5162.....	412
PXIe-5185/5186.....	449
PXIe-5622.....	468
PXI-5900.....	506
PXI/PCI-5922.....	509
NI Reconfigurable Oscilloscope Devices Help.....	541
Integration and System Considerations.....	583
PXI/PXIe Systems.....	583
PCI Systems.....	591
NI-SCOPE Soft Front Panel and InstrumentStudio.....	592
Related Documentation.....	593
NI-SCOPE 32-Bit Soft Front Panel Help.....	593
Programming.....	607
Programming Fundamentals.....	608
Array Measurements in NI-SCOPE.....	657
Scalar Measurements in NI-SCOPE.....	679
NI-SCOPE Instrument Driver FPGA Extensions.....	704
Reference.....	706
Glossary.....	3771
NI Services.....	3784



NI High-Speed Digitizers Help

May 2020, 370592AH-01

This help file contains technical and programming support for the NI-SCOPE instrument driver and the following NI high-speed digitizers and accessories:

- PXIe/PXI/PCI-5105
- PXIe-5110/5111/5113
- PXIe/PXI/PCI-5114
- PXIe/PXI/PCI-5122
- PXI/PCI-5124
- USB-5132/5133
- PXI/PCI-5142
- PXI/PCI-5152
- PXI/PCI-5153
- PXI/PCI-5154
- PXIe-5160/5162
- PXIe-5163
- PXIe-5164
- PXIe-5170/5171
- PXIe-5172
- PXIe-5185/5186
- PXIe-5622
- PXI-5900
- PXI/PCI-5922

The [Programming](#) section contains basic programming information and a tutorial for using NI-SCOPE to program digitizers in LabVIEW, LabWindows™/CVI™, Microsoft Visual C++, and Microsoft Visual Basic.

For LabVIEW reference material, refer to the [NI-SCOPE LabVIEW Reference Help](#). For C/C++ reference material, refer to the [NI-SCOPE Function Reference Help](#).

For more information about this help file, refer to the following topics:

[Related Documentation](#)

[Glossary](#)

[Technical Support and Professional Services](#)

© 2003–2020 National Instruments Corporation. All rights reserved.

Refer to the <National Instruments>_Legal Information directory for information about NI copyright, patents, trademarks, warranties, product warnings, and export compliance.

Related Documentation

Most manuals are available on the web in both HTML and PDF formats. You must have Adobe Reader installed to view the PDFs. Refer to the Adobe Systems Incorporated website at www.adobe.com to download the latest version of Adobe Reader. Refer to the National Instruments Product Manuals Library at ni.com/manuals for updated documentation resources.

The following documents contain information that you may find helpful as you use this help file:

- **NI-SCOPE Readme**
- **Maintain Forced-Air Cooling Note to Users**

Refer to the NI Product Manuals Library at ni.com/manuals for updated documentation resources.

Model Number	Documentation
--------------	---------------

5105	<ul style="list-style-type: none"> ▪ PXIe-5105 Specifications ▪ PXI-5105 Specifications ▪ PCI-5105 Specifications ▪ NI PXI/PXIe/PCI-5105 Getting Started Guide ▪ NI PXI/PXIe/PCI-5105 Calibration Procedure
5110 5111 5113	<ul style="list-style-type: none"> ▪ PXIe-5110 Specifications ▪ PXIe-5111 Specifications ▪ PXIe-5113 Specifications ▪ PXIe-5110/5111/5113 Getting Started Guide ▪ PXIe-5110 Calibration Procedure ▪ PXIe-5111 Calibration Procedure ▪ PXIe-5113 Calibration Procedure
5114	<ul style="list-style-type: none"> ▪ PXIe-5114 Specifications ▪ PXI-5114 Specifications ▪ PCI-5114 Specifications ▪ NI PXI/PXIe/PCI-5114 Getting Started Guide ▪ NI PXI/PXIe/PCI-5114 Calibration Procedure
5122	<ul style="list-style-type: none"> ▪ PXIe-5122 Specifications ▪ PXI-5122 Specifications ▪ PCI-5122 Specifications ▪ NI PXI/PXIe/PCI-5122 Getting Started Guide ▪ NI 5122/5142 Calibration Procedure
5124	<ul style="list-style-type: none"> ▪ PXI-5124 Specifications ▪ PCI-5124 Specifications

	<ul style="list-style-type: none"> ▪ NI PXI/PCI-5124 Getting Started Guide ▪ NI 5122/5142 Calibration Procedure
5132 5133	<ul style="list-style-type: none"> ▪ NI USB-5132 Specifications ▪ NI USB-5133 Specifications ▪ NI USB-5132/5133 Getting Started Guide ▪ NI 5132/5133 Calibration Procedure
5142	<ul style="list-style-type: none"> ▪ PXI-5142 Specifications ▪ PCI-5142 Specifications ▪ NI 5122/5142 Calibration Procedure
5152	<ul style="list-style-type: none"> ▪ PXI-5152 Specifications ▪ PCI-5152 Specifications ▪ PXI/PCI-5152 Getting Started Guide ▪ NI 5152/5153/5154 Calibration Procedure
5153	<ul style="list-style-type: none"> ▪ PXI-5153 Specifications ▪ PCI-5153 Specifications ▪ PXI/PCI-5153 Getting Started Guide ▪ NI 5152/5153/5154 Calibration Procedure
5154	<ul style="list-style-type: none"> ▪ PXI-5154 Specifications ▪ PCI-5154 Specifications ▪ PXI/PCI-5154 Getting Started Guide ▪ NI 5152/5153/5154 Calibration Procedure
5160 5162	<ul style="list-style-type: none"> ▪ PXIe-5160 Specifications ▪ PXIe-5162 Specifications ▪ NI PXIe-5160/5162 Getting Started Guide

	<ul style="list-style-type: none"> ▪ PXIe-5160 Calibration Procedure ▪ PXIe-5162 Calibration Procedure
5163	<ul style="list-style-type: none"> ▪ PXIe-5163 Specifications ▪ PXIe-5163 Getting Started Guide ▪ PXIe-5163 Calibration Procedure
5164	<ul style="list-style-type: none"> ▪ PXIe-5164 Specifications ▪ PXIe-5164 Getting Started Guide ▪ PXIe-5164 Calibration Procedure
5170 5171	<ul style="list-style-type: none"> ▪ PXIe-5170 Specifications ▪ PXIe-5171 Specifications ▪ PXIe-5170/5171 Getting Started Guide ▪ NI PXIe-5170R Calibration Procedure ▪ NI PXIe-5171R Calibration Procedure
5172	<ul style="list-style-type: none"> ▪ PXIe-5172 Specifications ▪ PXIe-5172 Getting Started Guide ▪ PXIe-5172 Calibration Procedure
5185 5186	<ul style="list-style-type: none"> ▪ NI PXIe-5185 Specifications ▪ NI PXIe-5186 Specifications ▪ NI High-Speed Digitizers Getting Started Guide ▪ PXIe-5185/5186 Calibration Procedure
5622	<ul style="list-style-type: none"> ▪ PXIe-5622 Specifications ▪ NI PXIe-5622 Getting Started Guide ▪ NI 5622 Calibration Procedure
5900	<ul style="list-style-type: none"> ▪ NI PXI-5900 Specifications

	<ul style="list-style-type: none"> ▪ NI High-Speed Digitizers Getting Started Guide ▪ NI 5900 Calibration Procedure
5922	<ul style="list-style-type: none"> ▪ PXI-5922 Specifications ▪ PCI-5922 Specifications ▪ PXI/PCI-5922 Getting Started Guide ▪ NI PXI/PCI-5922 Calibration Procedure

Getting Started

To get started using your NI-SCOPE device and driver software, refer to the getting started guide for your device. The getting started guide explains how to complete the following tasks:

- Install the software and hardware
- Configure the hardware in MAX
- Program the hardware
- Self-calibrate the hardware
- Make a first measurement
- Troubleshoot

The getting started guide for your device is printed in your hardware kit, installed with the driver software, and available at ni.com/manuals.



Tip Complete the getting started guide for your device before using the information in this help file.

Related Topics

[Integration and System Considerations](#)

[NI-SCOPE Examples](#)

Operating System Support

For information about the supported operating system (OS) for your device, refer to the **NI-SCOPE Readme** at Start»All Programs»National Instruments»NI-SCOPE»Documentation.

Creating an Application

You can begin programming with the following application development environments (ADEs):

- [LabVIEW 7.1 or later](#)
- [LabWindows/CVI 7.0 or later](#)
- [Microsoft Visual C++ \(MSVC\) 6.0 or later](#)
- [Microsoft Visual Basic 6.0 or later](#)
- [Microsoft Visual C# or Visual Basic .NET](#)

Each section includes basic information on how to create an application and how to find and use examples for each ADE.

Creating an Application with LabVIEW

To begin programming in LabVIEW, complete the following steps:

1. Launch LabVIEW.
2. Open an existing LabVIEW VI or create a new VI.
3. Make sure you are on the block diagram screen.
4. Open the NI-SCOPE palette. In LabVIEW 7.1 and later, the path to the palette depends on the current palette view in Tools»Options. In the Express palette view (default) the path is Input»Instrument Drivers»NI SCOPE.
5. Build your application using the appropriate [programming flow](#) steps.

LabVIEW Examples

You can find shortcuts to the LabVIEW examples at **Start»All Programs»National Instruments»NI-SCOPE»Examples**. These examples are also available in the NI-SCOPE portion of the LabVIEW functions palette.

Several forms of help are available for these examples. You can find an overall description of the purpose of the example by going to **File»VI Properties** and selecting **Documentation** from the Category listbox. For help on individual front-panel controls, activate the LabVIEW help window by selecting **Help»Show Context Help**, and placing the cursor over the control you want more information about. The block diagrams of the examples are documented with step-by-step descriptions of what the example does.

NI Example Finder

LabVIEW 7.0 or later users can use the NI Example Finder to search or browse examples. NI-SCOPE examples are classified by keyword, so you can search for a particular device or measurement function. To browse for NI-SCOPE examples, launch LabVIEW, select **Help»Find Examples**, then navigate to **Hardware Input and Output»Modular Instruments» NI-SCOPE (High-Speed Digitizers)**.

LabVIEW Real-Time Module

You can use the NI-SCOPE VIs on a LabVIEW Real-Time execution target. To develop an NI-SCOPE application in the LabVIEW Real-Time Module, follow the same steps used for developing any application in the LabVIEW Real-Time Module, with the addition of using NI-SCOPE VIs, as follows:

1. From the Function Palette, locate the NI-SCOPE VIs at **Instrument I/O Instrument Drivers»NI-SCOPE**.
2. Select the VIs that you want to use and drop them on the block diagram to build your application.

Supported LabVIEW Versions

LabVIEW Real-Time Module 7.1.1 or later

Supported Hardware Modules

Most NI high-speed digitizers are supported under the LabVIEW Real-Time Module.

The following devices are not supported under the LabVIEW Real-Time Module:

- PXIe-5110/5111/5113
- PXIe-5163
- PXIe-5164
- PXIe-5170/5171
- PXIe-5172

Unsupported Features

When using NI high-speed digitizers with the LabVIEW Real-Time Module, the NI-SCOPE Soft Front Panel is not supported.

Example Programs

You can use the LabVIEW Example Finder to search or browse through examples. NI-SCOPE examples are classified by keyword, so you can search for a particular device or measurement function. You can reach the NI-SCOPE examples from LabVIEW Example Finder (Help»Find Examples). Click the Browse tab, browse according to Task, and navigate to Hardware Input and Output/Modular Instruments/NI-SCOPE (High-Speed Digitizers). You can use all of the NI-SCOPE examples on a LabVIEW Real-Time execution target provided that the example is not specific to an unsupported hardware module.

Related Documentation

- For configuration instructions for remote systems, refer to the **Remote Systems Help** in Measurement & Automation Explorer (MAX) by selecting Help»Help Topics»Remote Systems in MAX.
- For more information on the LabVIEW Real-Time Module, refer to the **LabVIEW Real-Time Module User Manual** at ni.com/manuals.

Creating an Application with LabWindows™/CVI™

To begin programming an NI-SCOPE application with LabWindows/CVI, complete the following steps:

1. Launch LabWindows/CVI.
2. Open an existing or new source file (.c).
3. Include the NI-SCOPE header file, `niscope.h`, in your source code files (`#include "niscope.h"`).
4. Go to Instrument»Load, and select `niscope.fp`. You can find the installed location of this file in the **NI-SCOPE Readme**.
5. Add the source file to the project.
6. Build your application using the appropriate [programming flow](#) steps.

LabWindows/CVI Examples

Open the project file for the example you want to run. You can find shortcuts to the project files at Start»All Programs»National Instruments»NI-SCOPE»Examples. The source code for these examples is heavily documented, and all input and output values are documented to facilitate changing the code to perform different acquisitions. Each example also has a Readme file that provides an overall description of the example.

To find documentation for functions used in the examples, complete the following steps:

1. Double-click `niScope.fp`.
2. Select the function or function class you need more information about.
3. Click Help.

NI Example Finder

LabWindows/CVI 7.0 or later users can use the NI Example Finder to search or browse examples. NI-SCOPE examples are classified by keyword, so you can search for a particular device or measurement function. To browse for NI-SCOPE examples,

launch LabWindows/CVI, select Help»Find Examples, then navigate to Hardware Input and Output»Modular Instruments» NI-SCOPE (High-Speed Digitizers).

Creating an Application with Microsoft Visual C and C++

This topic assumes that you are using the Microsoft Visual C or C++ ADE to manage your code development and that you are familiar with the following ADEs:

- [Microsoft Visual C++ 6.0](#)
- [Microsoft Visual Studio 2010](#)

Creating an Application with Microsoft Visual C++ 6.0

To develop an NI-SCOPE application with Microsoft Visual C++ 6.0, follow these general steps:

1. Open an existing or new Visual C/C++ project to manage your application code.
2. Create source code files of type `.c` (C) or `.cpp` (C++).
3. Add the source code files to the project.
4. Add the following code in the source code files:


```
#include "nscope.h"
```
5. Add the NI-SCOPE include and library files to the project. `<IVIROOTDIR32>` is an alias to a specific National Instruments file folder location. Refer to the **NI-SCOPE Readme** for more information about installed file locations.



Tip You can modify an NI-SCOPE C example to create an application, and all required include and library files are added to the project for you. [Action: Verify that your driver includes C examples that include Visual C++ project files. If it does not, delete this tip.]

1. Select Project»Settings»C/C++»Preprocessor»Additional include directories, and add the paths, separated by a semicolon, to the `nscope.h` and `visa.h` files. The `nscope.h` and `visa.h` files are located in the `<IVIROOTDIR32>\Include` and `<VXIPNPPATH>\WinNT\Visa\include` directories, respectively.

2. Select **Project»Link»General»Object/Library Modules**, and add `niscope.lib`.
 3. Select **Project»Link»Input»Additional library path**, and add the path to the `niscope.lib` file. The `niscope.lib` file is located in the `<IVIROOTDIR32>\Lib\msc` or the `<IVIROOTDIR32>\Lib x64\msc` directory.
6. Build your application using the appropriate [programming flow](#) steps.

Creating an Application with Microsoft Visual Studio 2010

To develop an NI-SCOPE application with Microsoft Visual Studio 2010, follow these general steps:

1. Open an existing or new Visual C/C++ project to manage your application code.
2. Create source code files of type `.c` (C) or `.cpp` (C++).
3. Add the source code files to the project.
4. Add the following code in the source code files:


```
#include "niscope.h"
```
5. Add the NI-SCOPE include and library files to the project. `<IVIROOTDIR32>` is an alias to a specific National Instruments file folder location. Refer to the **NI-SCOPE Readme** for more information about installed file locations.



Tip You can modify an NI-SCOPE C example to create an application, and all required include and library files are added to the project for you. [Action: Verify that your driver includes C examples that include Visual C++ project files. If it does not, delete this tip.]

1. Select **Project»Properties»Configuration Properties»C/C++»General»Additional Include Directories**, add the paths, separated by a semicolon, to the `niscope.h` and `visa.h` files. The `niscope.h` and `visa.h` files are located in the `<IVIROOTDIR32>\Include` and `<VXIPNPPATH>\WinNT\Visa\include` directories, respectively.
2. Select **Linker»Input»Additional Dependencies**, and add `niscope.lib`.

3. Select **Linker»General»Additional Dependencies**, and add the path to the `niscope.lib` file. The `niscope.lib` file is located in the `<IVIROOTDIR32>\Lib\msc` or the `<IVIROOTDIR32>\Lib x64\msc` directory.

6. Build your application using the appropriate [programming flow](#) steps.

Microsoft Visual C and C++ Examples

Two sets of examples for Visual C/C++ exist. You can find the installed location of these examples in the **NI-SCOPE Readme**.

- The examples in the first set are C-based console applications that illustrate the NI-SCOPE function calls but have no provision for data display.
- The examples in the second set are C++ examples. They use Microsoft Foundation Classes (MFC) along with Measurement Studio tools for Visual C++. To compile these examples, you need these tools. The source code for these examples is documented, so you can change the code to fit your needs.

Microsoft Visual C Examples

The C examples were built and tested using Microsoft Visual C++ 6.0 with Service Pack 5.0. These examples are console-based with no graphical interface.

To build the examples in Microsoft Visual C++ 6.0 using the Microsoft NMAKE utility, complete the following steps:

1. Go to the directory for the particular example you want to use.
2. Run the `VCVARS32.BAT` batch file (located in the `\bin` directory of the MSVC compiler) to set up the environment variables for command line usage if they are not already set. You might need to increase the initial environment size of the DOS box to accommodate the added environment variables.
3. To build an example, run `nmake exemplename.mak`. The executable is built to the debug subdirectory by default.

To build the examples in Microsoft Visual C++ 6.0 using the Microsoft Developer Studio workspace, complete the following steps:

1. Go to the directory for the example you want to use.

2. Open the project workspace file (`.dsw`) that launches Developer Studio. You can find the installed location of this file in the **NI-SCOPE Readme**.
3. Add the `include` directory and `lib\msc` directories to the project. You can find the installed location of these examples in the **NI-SCOPE Readme**. To build the examples, the `INCLUDE` and `LIB` search paths might need to be modified to find `niscscope.h` and `niscscope.lib` (Tools»Options»Directories or Project»Settings).
4. Build the example.

Creating an Application with Visual Basic

To develop an NI-SCOPE application with Visual Basic, complete the following steps:

1. Open an existing or new project (`.vbproj`).
2. Find shortcuts to the examples at Start»All Programs»National Instruments»NI-SCOPE»Examples.
3. Go to Projects»References and select NI-SCOPE Library.
4. Build your application using the appropriate [programming flow](#) steps.

Microsoft Visual Basic Examples

The Visual Basic examples use an evaluation version of the NI Measurement Studio ActiveX UI controls. The examples were built and tested using Microsoft Visual Basic 6.0. For each example, there is a project file (`.vbproj`) that launches Developer Studio.



Note The evaluation version of the Measurement Studio UI controls is limited to 5 minutes of continuous use.

If you have problems getting the examples to load or run, complete the following steps:

1. Make sure the NI-SCOPE reference is loaded (go to Project»References, and load `\bin\niscscope_32.dll`. You can find the installed location of this file in the **NI-SCOPE Readme**).

2. Make sure the National Instruments Measurement Studio UI controls are loaded (go to Project»Components, and load `cwui.ocx` from the Windows system directory).
3. Install Service Pack 5 for Microsoft Visual Studio.

The source code for these examples is heavily documented, and all input and output values are documented to facilitate changing the code to perform different acquisitions.

Creating an Application Using Visual C# or Visual Basic .NET

To develop an NI-SCOPE application with Visual C# or Visual Basic .NET, complete the following steps:

1. Make sure you have installed the following on your computer:
 - Microsoft .NET Framework 2.0
 - NI-SCOPE 3.2 or later
2. Download the NI-SCOPE .NET Libraries.
 1. Go to ni.com/downloads.
 2. Click Drivers and Updates.
 3. Use the Search field to locate the latest version of the NI-SCOPE .NET Libraries.
 4. Click the Readme link for installation instructions and system requirements.

Forms of Device Synchronization

The following forms of synchronization are available, listed in order by synchronization accuracy.



Notes

- Synchronization accuracy specifications depend on the system and device(s) being used. Refer to your device **Specifications** for more information.

- Multi-instrument NI-SCOPE sessions automatically synchronize all channels of the included instruments, simplifying your test program development and maintenance. You can only create an NI-SCOPE session with multiple instruments of the same model and in the same chassis.
- Sharing a clock (reference clock or sample clock) among devices can achieve time locking. In some cases, time locking among clocks can achieve greater determinism; however, it does not affect any overall delays in signals sent.

Software-Based Synchronization

You can send a synchronization software command from a host computer to a device. Synchronizing events by software is not deterministic on operating systems such as Windows or Mac.

- **Accuracy:** tens of milliseconds

Related Topics

[Supported Synchronization Methods by Device](#)

Time-Based Synchronization



Note You can use a time-based protocol, such as GPS, 1588, or IRIG-B, to coordinate events at large distances, typically greater than 10 m.

- **Accuracy:** under 100 nanoseconds

Related Topics

[Introduction to NI Clock Disciplining for PXI Express](#)

[PXI_Clk10 Disciplining and Synchronization across Multiple PXIe Chassis](#)

[Timing and Synchronization Systems](#)

[Synchronization Products](#)

Signal-Based Synchronization

PXI Trigger or RTSI Trigger Routing (SCOPE, FGEN, HSDIO, RF)

You can send an electrical signal from one device (master) to another (slave) to signal an event using the trigger routing available from a PXI chassis or RTSI cable. The devices dictate the level of determinism. The signal path length mainly determines the synchronization accuracy.

- Accuracy: tens of nanoseconds

Related Topics

[Synchronization Explained](#) (trigger routing and sharing clocks and triggers)

[Triggers Offered by PXI Standard](#)

[PXI Star Trigger Line](#)

External Triggering



Note In this case, **external** refers to triggering external to your PXI chassis or external to the device for other device form factors, for example USB and PCIe.

You can send triggers to each device through PFI, channel, or other I/O lines on the device front panel. When trigger lines have matched signal path lengths, measured synchronization results may improve.

- Use time locking for greater determinism.
- Accuracy: ± 1 sample clock

Related Topics

[Improving System Clock Error](#)

[When to Use External Sample Clock](#)

NI-TClk (SCOPE, FGEN, HSDIO, RF)

You can use NI-TClk to programmatically phase align clocks and coordinate triggers between devices and multiple chassis. NI-TClk is an advanced synchronization method, and is not supported by all devices.

- Use time locking for greater determinism.
- Accuracy: 10-500 ps



Note NI-SCOPE does not support using NI-TClk to synchronize multi-instrument NI-SCOPE sessions with other NI-SCOPE sessions or other driver sessions.

Related Topics

[National Instruments T-Clock Technology for Timing and Synchronization of Modular Instruments](#)

[Supported Synchronization Methods by Device](#)

Supported Synchronization Methods by Device

Device Name	Software-based Synchronization	PXI Trigger or RTSI Trigger Routing	External Triggering	NI-TClk (Advanced)
PXIe/PXI/PCI-5105	✓	✓	✓	✓
PXIe-5110/5111/5113	✓	✓	✓	✓
PXIe/PXI/PCI-5114	✓	✓	✓	✓
PXIe/PXI/PCI-5122	✓	✓	✓	✓
PXI/PCI-5124	✓	✓	✓	✓
USB-5132/5133	✓	✗	✓	✗
PXI/PCI-5142	✓	✓	✓	✓
PXI/PCI-5152	✓	✓	✓	✓
PXI/PCI-5153	✓	✓	✓	✓
PXI/PCI-5154	✓	✓	✓	✓
PXIe-5160/5162	✓	✓	✓	✓
PXIe-5163	✓	✓	✓	✓
PXIe-5164	✓	✓	✓	✓

PXIe-5170/5171	✓	✓	✓	✓
PXIe-5185/5186	✓	✓	✓	✓
PXIe-5622	✓	✓	✓	✓
PXI/PCI-5922	✓	✓	✓	✓

NI-SCOPE Tutorial

Example Programs

NI-SCOPE ships with several examples that demonstrate some basic digitizer applications. You can find these examples at [Start»All Programs»National Instruments»NI-SCOPE»Examples](#).



Note The NI-SCOPE examples assume that you are already familiar with your ADE. If you are unfamiliar with your ADE, consult an introductory text on the ADE before you begin the [Getting Started](#) example.

NI-SCOPE provides the same functionality in two different formats—as virtual instruments (VIs) in LabVIEW and as functions in traditional programming languages.

Basic Steps

The tutorial explains how to do the following:

1. [Initialize the session](#).
2. [Configure an application](#).
3. [Acquire data](#).
4. [Retrieve error information](#).
5. [Close the session](#).

After you finish the NI-SCOPE Tutorial, you can find more advanced information in the following sections:

- [Acquisition Functions](#)
- [Configuration Functions](#)

- Attributes and Attribute Functions

Instrument Drivers

An instrument driver is a set of software routines that control programmable instruments. Each routine corresponds to a programmatic operation such as configuring, reading from, writing to, and triggering the instrument. Instrument drivers simplify instrument control and reduce test program development time by eliminating the need to learn the programming protocol for each instrument. NI-SCOPE, NI-FGEN, NI-DMM, and NI-HSDIO are examples of instrument drivers provided by National Instruments.

Instrument Driver Structure

All instrument drivers have the same basic hierarchy. Instrument drivers are written from low-level components that are used to program instruments. These component functions fit into six categories—Initialize, Configuration, Action/Status, Data, Utility, and Close.

Initialize

All instrument drivers have an Initialize function. It is the first instrument driver function called and establishes communication with instruments. Optionally, it can also perform instrument identification query and reset operations. Initialization also places instruments in a specific state, usually the default power on state.

Configuration

Configuration functions are collections of software routines that configure instruments to perform a desired operation. The number of Configuration functions depends on the complexity of the instruments. After the Configuration functions are called, the instruments are ready to take measurements.

Action/Status

The Action/Status category contains two types of functions. Action functions cause instruments to initiate or terminate test and measurement operations such as arming the triggering system or generating a stimulus. Action functions differ from

Configuration functions in that they do not change instrument settings; instead, they order the instrument to carry out an action based on its current configuration settings. Status functions return the current status of instruments or of pending operations. Specific routines in this category and the actual operations they perform are at the discretion of the instrument driver developer, and are usually created as required by other functions.

Data

Data functions include calls to transfer data to or from instruments. Examples of Data function operations include reading a measured value or waveform from a measurement instrument, and downloading waveforms or digital patterns to a source instrument. Specific routines in this category depend on the instrument.

Utility

Utility functions perform a variety of operations auxiliary to the most-used instrument driver calls. These functions include the majority of the template instrument driver calls such as reset, self-test, revision, and error query, and may include other custom routines such as calibration or storing/recalling instrument configurations.

Close

All instrument drivers include a Close function. The Close function terminates the software connection to instruments and deallocates system resources used during that instrument session.

Initialize

Because you can have multiple digitizers connected to your computer, you must tell NI-SCOPE which digitizer(s) to communicate with by opening a session to the digitizer or digitizers with an Initialize function.

What Is a Session?

A session establishes a connection between digitizers and your application. After this connection is established, the digitizer(s) can transmit data to your application.

Sessions allow the driver to cache previous settings, which greatly improves performance.

How Do You Create a Session?

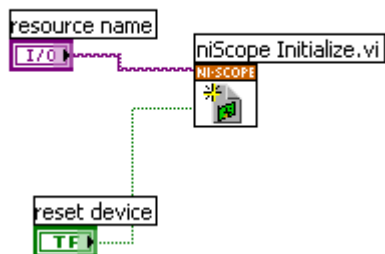
To create a session, use the `Initialize with Options` or `Initialize` functions. These functions return a handle that allows you to communicate with the digitizer(s) using other function calls in your application.

You can create sessions that include multiple instruments of the same model and in the same chassis. This automatically synchronizes all channels of the included instruments, simplifying your test program development and maintenance. To create a session to multiple digitizers, enter the instrument names as a comma-delimited list of strings in the form **PXI1Slot3,PXI1Slot4** for the Resource Name input parameter of an Initialize function.



Note You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes.

LabVIEW Example—Initializing a Session



C Example—Initializing a Session

```
niScope_init ("PXI1Slot2", VI_TRUE, VI_TRUE, &vi);
```



Notes

- When you run your application, you must use the resource name assigned by Measurement & Automation Explorer (MAX). Find or set the resource name for your digitizers by launching MAX and selecting **Devices and Interfaces**.

- Setting the ID Query parameter to `TRUE` has no effect with NI-SCOPE. NI-SCOPE automatically verifies that the digitizer(s) you initialize are supported—therefore, setting this parameter to `TRUE` is not necessary.
- Setting the Reset Device parameter to `TRUE` resets digitizers during initialization.



Caution Resetting digitizers may cause wear on the relays, so you should reset only when necessary.

Configure the Application

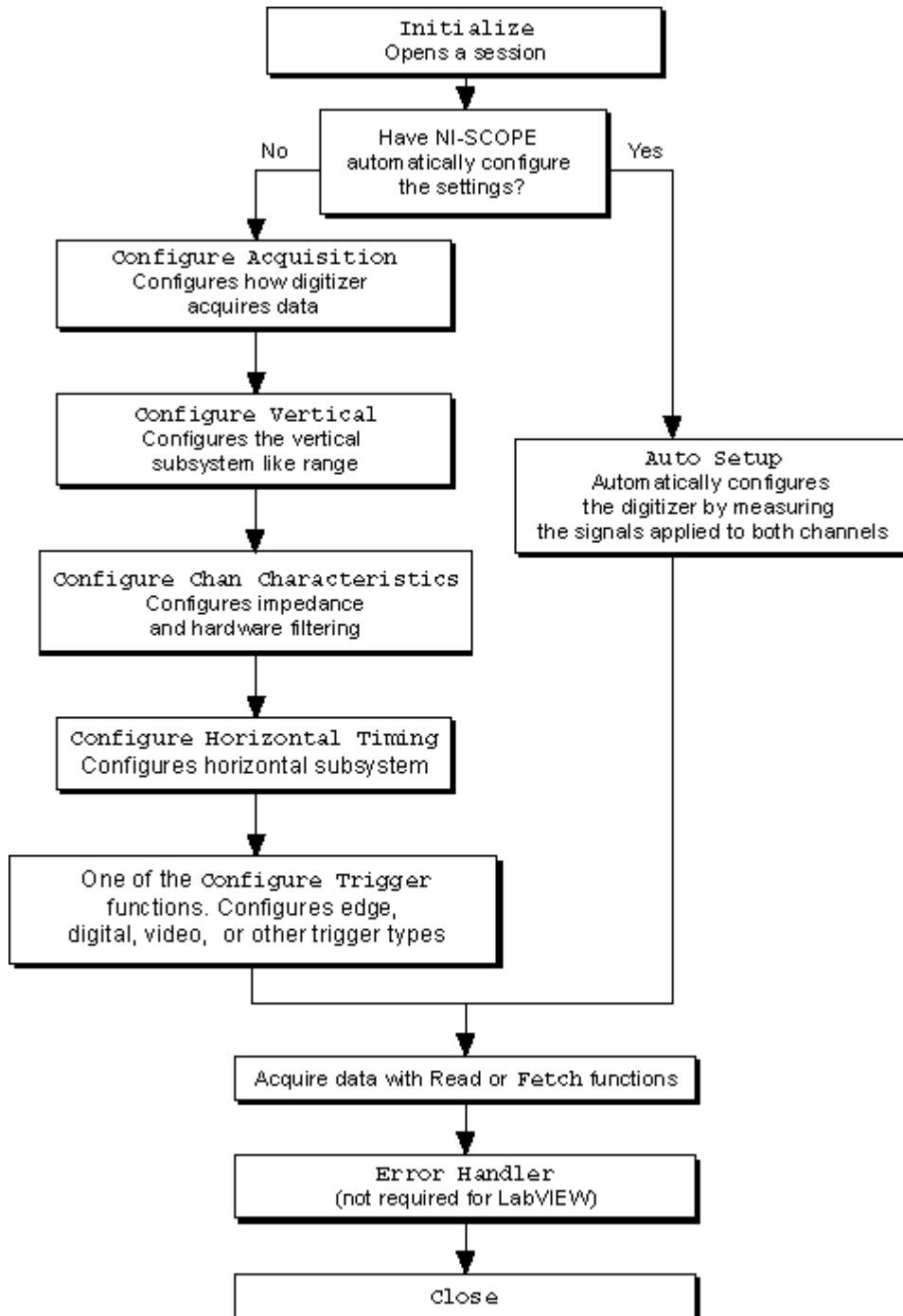
After you have opened a session to the digitizer(s) with an Initialize function, you need to configure your application. For example, you might want to set input parameters (also known as controls and constants in LabVIEW) such as the vertical range, minimum sample rate, and triggering options.

You can use one of the following two methods to configure an application:

- [Using Auto Setup](#)—This method automatically configures input parameters.
- [Using Configuration Functions](#)—This method allows you to set the input parameters yourself.

Programming Flow

The following diagram shows the basic programming flow for using either Auto Setup or configuration functions.

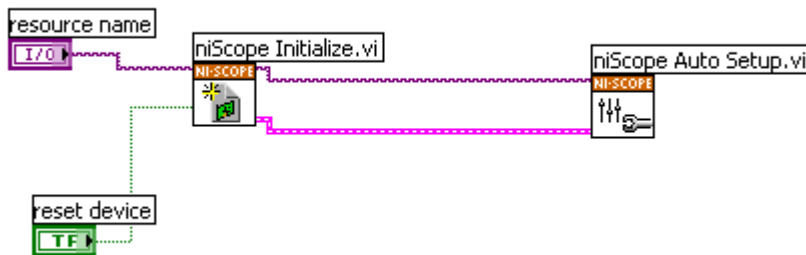


Configuring an Acquisition Using Auto_Setup

After you have opened a session to the digitizer(s), you need to configure your application. For the `Getting Started` example, we use the `Auto_Setup`

function to configure the digitizer. The `Auto Setup` function senses the input signal and automatically configures many of the instrument settings, such as the vertical range, sample rate, minimum record length, and trigger levels.

LabVIEW Example—Configuring An Acquisition



C Example—Configuring An Acquisition

```
niScope_AutoSetup (vi);
```

Alternative Configuration Method—Using the Configuration Functions to Set Up an Acquisition

If you want to customize sample rates, change triggers, or alter the offset, you must use the Configuration functions such as `Configure Vertical` and `Configure Horizontal Timing` instead of the `Auto Setup` function.

Refer to [Configuring an Acquisition Using Configuration Functions](#) for more information on using the Configuration functions.

Configuring an Acquisition Using Configuration Functions

As an alternative to using the `Auto Setup` function, you can use the Configuration functions to set up an acquisition in which you want to set triggers, input impedance, DC offset, vertical range, sampling rate, and so on.

Configuration Example

The `Configured Acquisition` example demonstrates using the Configuration functions. This example supports most of the configuration functions of NI-SCOPE, so you can experiment with the digitizer.

Related Topics

[Configuring the Acquisition Type](#)

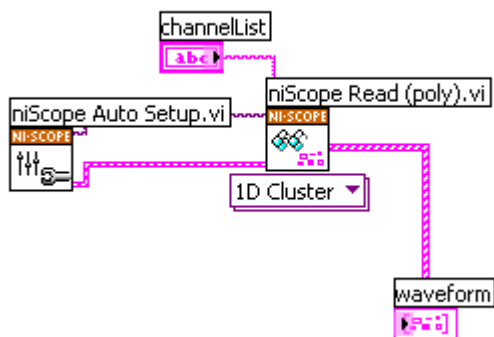
[Configuring the Vertical and Channel Settings](#)

[Configuring the Horizontal Settings](#)

Acquiring the Data

NI-SCOPE has two kinds of functions that acquire data—Read and Fetch functions. For simplicity, this example uses a Read function.

LabVIEW Example—Acquiring the Data



The channels input parameter specifies the channel or channels that NI-SCOPE acquires data from.

C Example—Acquiring the Data

This example includes additional functions and parameters not needed in LabVIEW. You need to call `niScope_ActualRecordLength` to get the actual record length that NI-SCOPE uses so you can create enough space for the waveform array.

You can then initiate the acquisition and fetch the data with `niScope_Read`. After fetching the data, you need to pass a pointer to extract and plot the acquired data. The method you use to do this varies depending on your ADE.

```
// Get the actual number of samples to be acquired
niScope_ActualRecordLength (vi, &actualRecordLength);

// Get the number of waveforms available for the
// given channelList
niScope_ActualNumWfms (vi,channelList, &numWfm);

// Allocate space for the waveform information
wfmInfoPtr = malloc (sizeof (struct niScope_wfmInfo) * num
Wfm);

// Allocate space for the waveform array
wfmPtr = malloc (sizeof (ViReal64) * actualRecordLength *
numWfm);

// Check if allocations succeeded
if (!wfmPtr || !wfmInfoPtr)

    return NISCOPE_ERROR_INSUFFICIENT_MEMORY;

// Read the data (Initiates an acquisition and fetches
// the data)
niScope_Read (vi, channelList, timeout, actualRecordLength
, wfmPtr, wfmInfoPtr);
```

About the Parameters

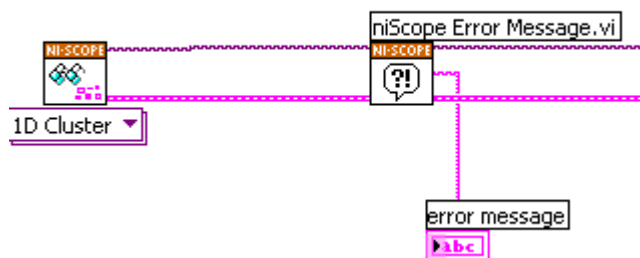
The `channelList` parameter specifies the channel or channels that NI-SCOPE acquires data from. The `wfmInfoPtr` parameter returns the values needed to display a waveform: **relative initial x** is the time of the first point in the waveform with respect to the trigger and is expressed in seconds; **absolute initial x** is the absolute timestamp of the first point in the waveform; and **x increment** is the length of time between points in the waveform in seconds. The `wfmPtr` parameter is a buffer into which NI-SCOPE stores the waveform it reads. The units for the individual array elements are volts.

Retrieving Error Information

LabVIEW

Make sure that you have wired the Error In and Error Out parameters for each VI in your application, so the VI (unless it is a Close, Abort, or Reset VI) does not execute if there is an error. The Error Description parameter displays messages for any errors encountered when running the application.

LabVIEW Example—Retrieving Error Information



Other ADEs

The `Error Handler` function translates error codes into explanations to help you debug your application.

C Example—Retrieving Error Information

```
// The handleErr macro checks the return value of a
// function and jumps to the Error: label
Error:
// Free all the allocated memory
if (wfmInfoPtr)
    free (wfmInfoPtr);
if (waveformPtr)
    free (waveformPtr);
// Display messages
if (error != VI_SUCCESS)
    niScope_errorHandler (vi, error, errorSource, errorMess
```

```

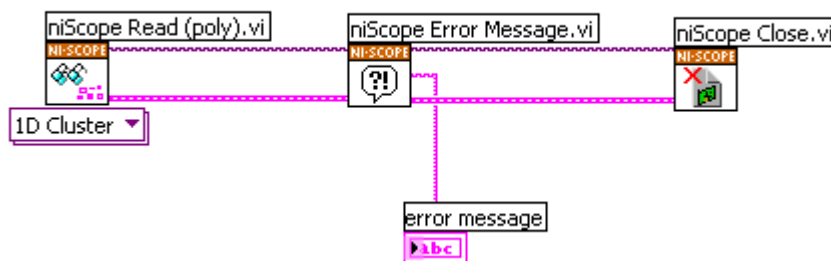
age);
// Interpret the error
else
    strcpy(errorMessage, "Acquisition successful!");
// The handleErr macro checks the return value

```

Closing the Session

The closing step closes the session and deallocates any resources the session used. Closing the session is important because it releases any temporary buffers that were created to transfer data between the digitizer and the host memory.

LabVIEW Example-Closing the Session



C Example—Closing the Session

```

// Close the session
if (vi)
    niScope_close(vi);

```

NI-SCOPE and InstrumentStudio

When you install NI-SCOPE on a 64-bit system, you can monitor, control, and record measurements from supported devices using InstrumentStudio. InstrumentStudio is a software-based front panel application that allows you to perform interactive measurements on several different device types, including oscilloscopes, in a single program.



Note InstrumentStudio support was first available in NI-SCOPE 18.1. InstrumentStudio is supported only on 64-bit systems. If you are using a 32-bit system, use the [NI-SCOPE Soft Front Panel](#).

InstrumentStudio is automatically installed when you install the NI-SCOPE driver. You can access InstrumentStudio in one of the following ways:

- From the Windows start menu, select National Instruments»NI-SCOPE Soft Front Panel. This launches InstrumentStudio and runs a soft front panel populated with NI-SCOPE devices.
- From the Windows start menu, select National Instruments»InstrumentStudio [**year**]. This launches InstrumentStudio and runs a soft front panel populated with devices detected on your system.
- From Measurement & Automation Explorer (MAX), select a device and then click Test Panels.... This launches InstrumentStudio and runs a soft front panel for the device you selected.

For more information, refer to [NI-SCOPE Soft Front Panel and InstrumentStudio](#).

NI-SCOPE Examples

NI-SCOPE includes several example applications that demonstrate the functionality of your device and can serve as interactive tools, programming models, and building blocks for your own applications.

NI Example Finder



The NI Example Finder is a utility that organizes examples into categories and allows you to browse and search installed examples. For example, search for "SCOPE" to locate all NI Digitizers signal generator examples. You can see descriptions and compatible hardware models for each example or see all the examples compatible with one particular hardware model.






To locate examples using the NI Example Finder within LabVIEW or LabWindows/CVI, select Help»Find Examples and navigate to Hardware Input and Output»Modular Instruments»NI-SCOPE.

Installed Example Locations

Software Application	Installed Example Locations
LabVIEW	<LabVIEW>\examples\instr\niScope, where <LabVIEW> is the LabVIEW directory for the specific LabVIEW version you installed on your system.
LabWindows/CVI	Program Files\IVI Foundation\IVI\Drivers\niScope\Examples\CVI
ANSI C or Microsoft Visual C/C++	<p><NIDocDir>\NI-SCOPE\examples, where <NIDocDir> is one of the following directories:</p> <ul style="list-style-type: none"> ▪ Windows 8/7/Vista—Users\Public\Documents\National Instruments ▪ Windows XP—Documents and Settings\All Users\Shared Documents\National Instruments

Common NI-SCOPE Examples

NI-SCOPE Example	Description
	<p>niScope EX Getting Started.vi</p> <p>This example opens a session to the NI-SCOPE driver and uses the Auto Setup VI to optimally configure the session properties and displays the acquired channel data.</p>
	<p>niScope EX Quick Start.vi</p> <p>This example opens and configures an NI-SCOPE session and allows you to experiment with numerous configurations, including acquisition types and triggering modes, since it supports nearly the entire functionality of NI-SCOPE. It is designed so that if the recommended input signal is used, most configurations will return valid results.</p>

	niScope EX Configured Acquisition.vi	This example configures all the digitizer's vertical, horizontal, and triggering properties before every acquisition. It allows you to experiment with numerous configurations, including acquisition types and triggering modes, since it supports nearly the entire functionality of NI-SCOPE.
	niScope EX Fetch Forever.vi	This example demonstrates using continuous acquisition to fetch an infinite record of data (until you press "Stop").
	niScope EX Stream to Disk.vi	This example demonstrates streaming a continuous acquisition to disk.
	niScope EX Advanced Measurement Library.vi	This example illustrates using some of the advanced measurement library functions such as waveform processing and filtering.
	niScope EX Multi-Device Configured Acquisition (TClk).vi	This example demonstrates how to easily synchronize an arbitrary number of digitizers with NI-TClk.

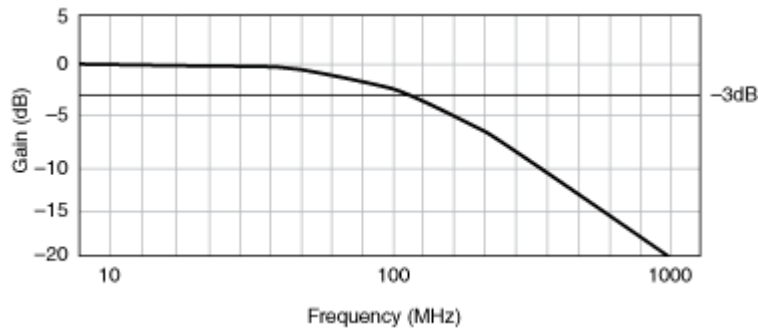
Fundamentals

Expand this book for more information on digitizer fundamentals.

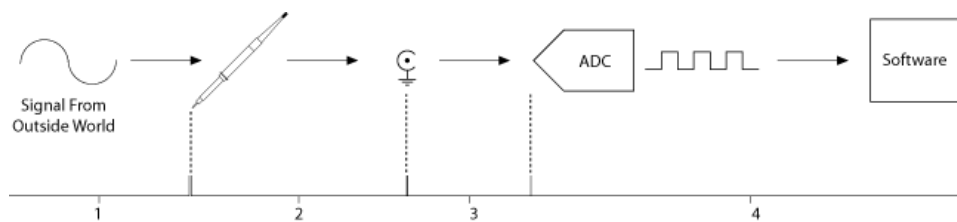
Bandwidth

Bandwidth describes the difference between limiting frequencies within which the input signal can pass through the system with minimal amplitude loss — from the input at the tip of the probe or test fixture to the output data. The limiting frequencies that determine the bandwidth include both a high and a low frequency that are specified as the frequency (in Hz) at which a sinusoidal input signal is

attenuated to 70.7% of its original amplitude. This point is known as the -3 dB point.



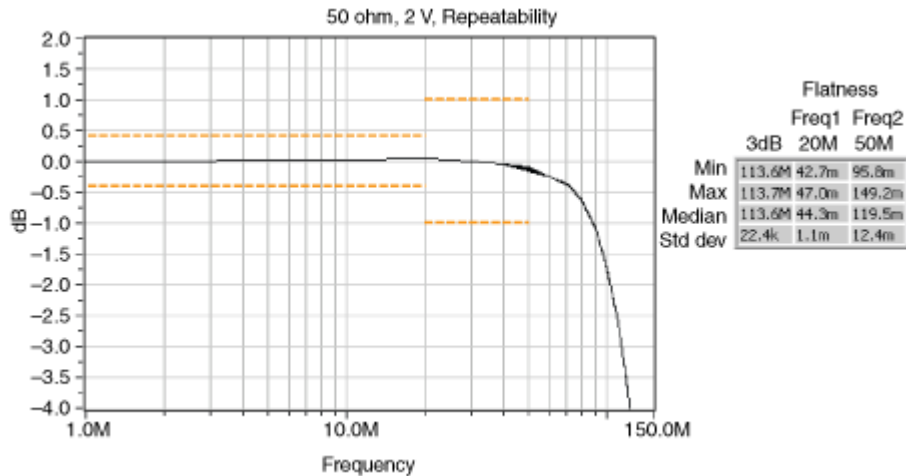
The following figure shows the four main areas that can affect the bandwidth of a digitizer.



1. Outside world to probe
2. Probe to input connector
3. Input connector to ADC
4. ADC to software

Flatness

Flatness is an effect on the acquired waveform that is frequency dependent. As the frequency rises, the amplitude slowly falls toward the 3 dB cutoff point of the bandwidth. When signals are composed entirely of frequencies below this cutoff point, the measured signal can appear slightly different than the input signal. The higher frequency components of the signal are attenuated more than the lower frequency components, changing the overall signal. Flatness describes how well the analog front end passes signals of different frequencies. A maximally flat front end passes all frequencies with the same amount of attenuation, so the measured signal looks like the input signal. However, in the real world, as the frequency rises, the measured input signal slowly falls toward the -3 dB point.



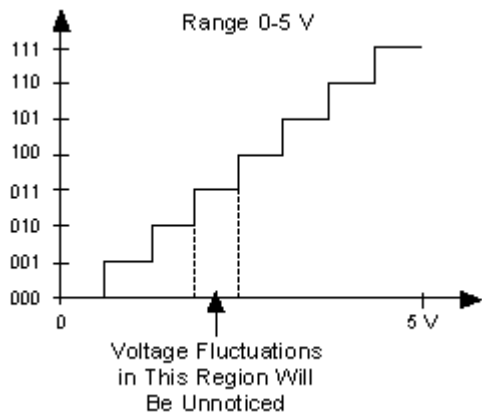
Resolution

Resolution is the smallest input voltage change a digitizer can capture. Resolution can be expressed in bits (LSB), in proportions, or in percent of full scale. For example, a system has 12-bit resolution, one part in 4,096 resolution, and 0.0244% of full scale.

Resolution limits the precision of a measurement. The higher the resolution (number of bits), the more precise the measurement. An 8-bit ADC divides the vertical range of the input amplifier into 256 discrete levels. With a vertical range of 10 V, the 8-bit ADC cannot ideally resolve voltage differences smaller than 39 mV. In comparison, a 14-bit ADC with 16,384 discrete levels can ideally resolve voltage differences as small as 610 μ V.

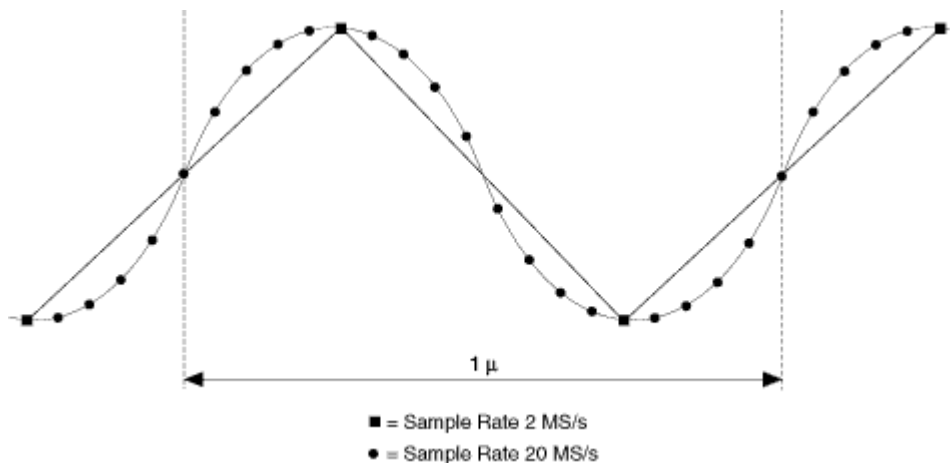
Bits	Resolution	%FS
8	1/256	.390
10	1/1024	.0976
12	1/4096	.0244
14	1/16384	.0061
16	1/65536	.0015

The following figure shows the transfer function of a 3-bit ADC with a vertical range of 5 V having an ideal resolution of 5/8 V/LSB.



Sample Rate

Sample rate, specified in samples per second (S/s), is the rate at which a signal is sampled and digitized by an analog-to-digital converter (ADC). According to the [Nyquist theorem](#), a sample rate at least twice the highest frequency of the signal produces accurate measurements if the [analog bandwidth](#) is wide enough to let the signal pass through without attenuation. A higher sample rate captures more waveform details for the time domain measurements. The following figure illustrates a 1 MHz sine wave sampled by a 2 MS/s ADC and a 20 MS/s ADC. The faster ADC digitizes 20 points per cycle of the input signal compared with 2 points per cycle with the slower ADC. In this example, the higher sample rate more accurately captures the waveform shape.



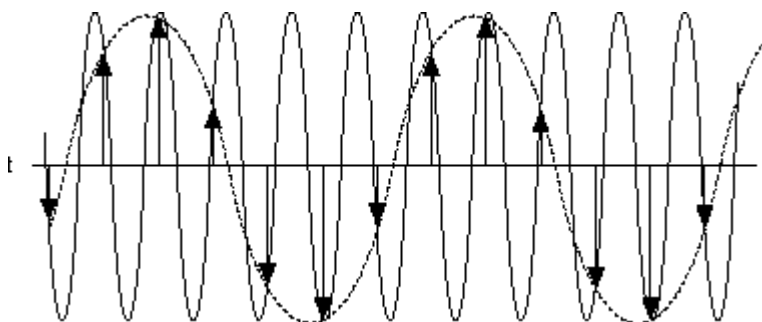
Related Topics

[Undersampling](#)

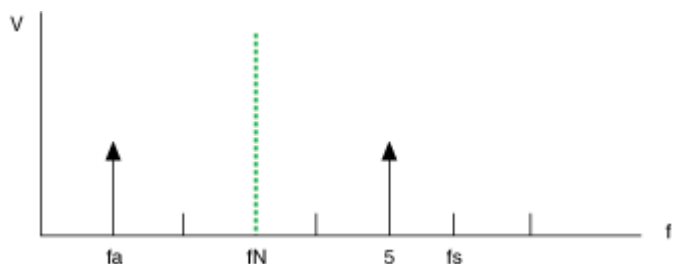
[Oversampling](#)

Nyquist Theorem

The Nyquist theorem states that a signal must be sampled at least twice as fast as the highest frequency component of the signal to accurately reconstruct the waveform. Frequency components higher than half the sample rate are mistakenly interpreted as low-frequency components, a phenomenon known as [aliasing](#). The following figure shows a 5 MHz sine wave digitized by a 6 MS/s ADC. The dotted line indicates the aliased signal recorded by the ADC at that sample rate. The 5 MHz frequency aliases back into the passband, falsely appearing as a 1 MHz sine wave. Inserting a lowpass filter with a cutoff at the Nyquist rate prevents aliasing in the passband.



The following figure shows the frequency view of the same example.



where

$f_s = 6 \text{ MS/s}$

$f_N = 3 \text{ MHz}$ (the Nyquist frequency)

$f_a = 1 \text{ MHz}$ ($1f_s - 5 \text{ MHz}$)

Related Topics

[Undersampling](#)

[Oversampling](#)

Aliasing

Aliasing is the misrepresentation of high frequencies as lower frequencies. An alias appears in sampled data acquired at too low a sampling rate compared to the [Nyquist](#) frequency of the signal being sampled.

In systems where you want to perform accurate measurements using sampled data, the sampling rate must be set high enough to prevent aliasing, or an optional anti-aliasing filter must be introduced before the A/D converter to restrict the bandwidth of the input signal to meet the sampling criteria. An anti-aliasing filter attenuates unwanted high-frequency signals (which otherwise would appear as undesired, aliased frequency components) of an analog signal prior to its conversion into a digital value. The actual bandwidth in which you can make correct measurements without aliasing is called the alias-free bandwidth. After aliasing has been introduced into a sampled signal, there is no general way to remove it.

In video applications, several forms of visual aliasing are possible. Temporal aliasing, where the wheel of a car seems to turn backwards, raster scan aliasing that results in twinkling on sharp horizontal lines, and stair-step aliasing, where jagged edges appear on lines running at an angle to the scan.

Oversampling

Oversampling is sampling at a rate beyond twice the highest frequency component of interest in the signal. Because real-world signals are not perfectly filtered and often contain frequency components greater than the [Nyquist](#) frequency, oversampling can be used to increase the [foldover frequency](#) so that these unwanted components of the signal do not alias into the passband. Oversampling is also necessary when trying to capture fast edges, transients, and one-time events.

Undersampling

Undersampling is essentially sampling too slowly, or sampling at a rate below the [Nyquist](#) frequency for a particular signal of interest. Undersampling leads to [aliasing](#) and the original signal cannot be properly reconstructed. However, undersampling also requires less memory, so it may be useful in certain applications.

Record Length

Record length refers to the number of samples, or data points, the digitizer acquires for each channel in a single acquisition.

To determine record length, call the [niScope Actual Record Length VI](#) or the [niScope ActualRecordLength](#) function. You can also obtain the record length using the [Actual Record Length](#) property or the `NISCOPE_ATTR_HORZ_RECORD_LENGTH` attribute.

Maximum Duration

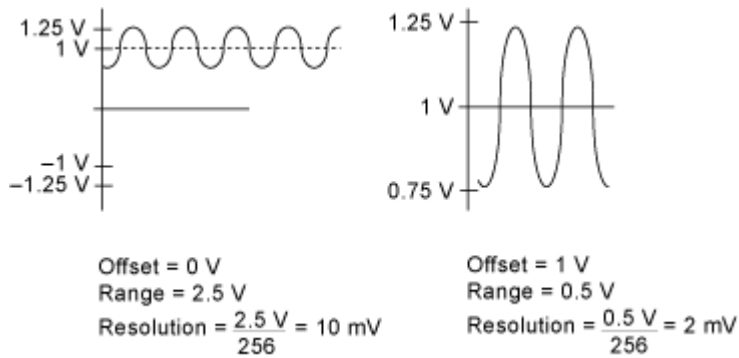
Record length limits the maximum duration of a single-shot acquisition. For example, with a 1,000-sample record and a sample rate of 20 MS/s, the duration of acquisition is 50 μ s (the number of points multiplied by the acquisition time per sample, or 1,000 x 50 ns). With a 100,000-sample record and a sample rate of 20 MS/s, the duration of the acquisition is 5 ms (100,000 x 50 ns).

Vertical Range and Offset

Vertical range is the peak-to-peak voltage span that a digitizer can measure at the input connector. Most digitizers have several choices for vertical range.

Vertical offset is the voltage the vertical range is centered on. Vertical offset positions the vertical range around an arbitrary DC value. Using this offset allows you to examine small changes in the input signal, which can improve the accuracy of your measurement.

For example, imagine that you are acquiring the waveform shown in the following figure that outputs 0.75 V to 1.25 V. Without using vertical offset, you would need to specify a range of 2.5 V (± 1.25 V) to capture the waveform.



Input Coupling

On many digitizers, you can configure the input channels to be DC-coupled, AC-coupled, or GND coupled. DC coupling allows DC and low-frequency components of a signal to pass through without attenuation. In contrast, AC coupling removes DC offsets and attenuates low frequency components of a signal. This feature can be exploited to zoom in on AC signals with large DC offsets, such as switching noise on a 12 V power supply. GND coupling disconnects the input and internally connects the channel to ground to provide a ground, zero-voltage reference.

Refer to the specifications for your specific digitizer for input limits that must be observed regardless of coupling.

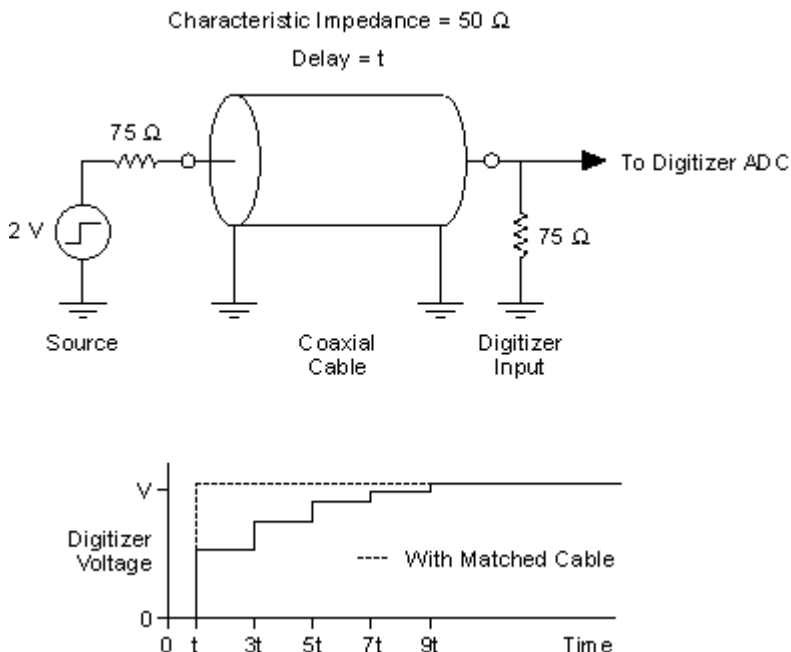
Impedance and Impedance Matching

When broadband signals are carried on transmission lines of any significant length, care must be taken that the transmission medium is matched to its terminations. The source and load impedances should equal the characteristic impedance of the transmission line, as this minimizes signal reflections. The presence of impedance discontinuities or mismatches degrades the amplitude and phase accuracy, as well as the temporal fidelity, of measurements made with a digitizer. The example below shows one of the most common mismatch errors encountered in such measurements.

Example

Selectable termination impedances are provided at the digitizer inputs to accommodate the most popular coaxial cable characteristic impedances: 50 Ω and

75 Ω. The diagram below illustrates what happens when a coaxial cable of the wrong characteristic impedance (50 Ω) is used with 75 Ω source and load impedances.



The pulse encounters impedance mismatches at each end of the cable, whereupon it is partially reflected. The reflected pulse traverses the cable back and forth numerous times, diminishing at each end by the reflection coefficient, Γ .

$$\Gamma = \frac{v_r}{v_i} = \frac{z_t - z_0}{z_t + z_0};$$

where

v_r = reflected voltage

v_i = incident voltage

z_t = termination impedance

z_0 = characteristic impedance

The resulting voltage waveform recorded by the digitizer is distorted by the asymptotic decay of the reflected pulse as shown, exaggerated for visual effect. Impedance discontinuities of smaller magnitude and/or duration have correspondingly smaller effects. Also displayed is the waveform that results when a cable of matched impedance (75 Ω) is used.

Mismatch Uncertainty

Impedance matching is also important for preserving the absolute power measurement accuracy of a digitizer. The accuracy with which power can be measured is limited by mismatch error. The mismatch error in a system can be shown to be bounded by:

$$\frac{(1 - |\Gamma_L|^2)}{(1 + |\Gamma_L| \cdot |\Gamma_G|)^2} \leq \text{mismatch error} \leq \frac{(1 - |\Gamma_L|^2)}{(1 - |\Gamma_L| \cdot |\Gamma_G|)^2}$$

where

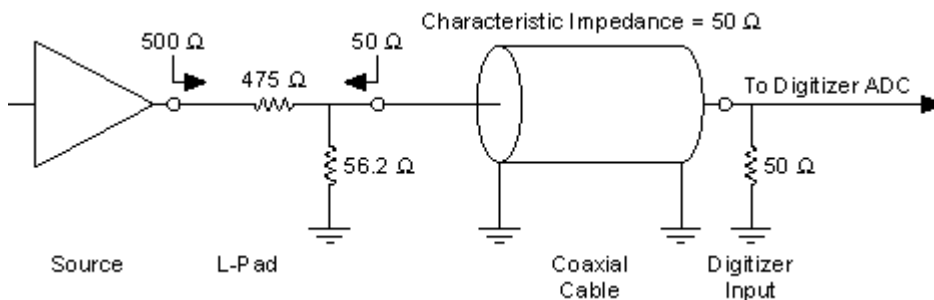
Γ_L = load reflection coefficient

Γ_G = generator reflection coefficient

The denominator term represents mismatch uncertainty, which is a fundamental limit to the power transfer accuracy that can be achieved across a mismatched junction.

Resistive Matching

Signal sources with low (high) source impedance can be matched with a resistor placed in series (shunt) such that the total source impedance (admittance) is matched to the cable characteristic impedance (admittance). Sources that are not capable of driving the cable impedance directly can be coupled through a matching L-pad, an example of which is shown in the following figure. In this case, the source sees a 500 Ω load, while the source impedance presented to the cable is 50 Ω . High-frequency components and layout techniques should be used throughout to minimize parasitic effects.

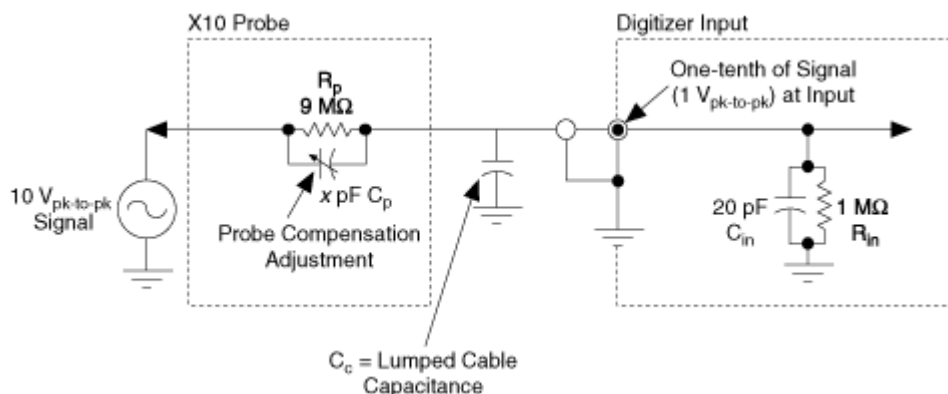


Probes and Their Effects

Probes work with digitizers as part of your measurement system. Signals travel from the tip of the probe to the input amplifier and are then digitized by the ADC. This signal path makes the probe an important electrical system component that can affect the accuracy of the measurement. A probe can potentially influence measured amplitude and phase, and the signal can pick up additional noise on its way to the input stage. Although NI high-speed digitizers do not ship with probes, several types of probes are available, including passive, active, and current probes.

Passive Probes

The passive probe is the most widely used general-purpose probe. Passive probes are specified by bandwidth (or rise time), attenuation ratio, compensation range, and mechanical design aspects. Probes with attenuation, X10, X100, or X1000, have a tunable capacitor that can reduce capacitive effects at the input. The ability to cancel or minimize effective capacitance improves the probe's bandwidth and rise time. The following figure shows a typical X10 probe model.



Adjust the tunable capacitor, C_p, to obtain a flat frequency response. C_p is the probe capacitance, R_p is the probe resistance, C_{in} is the input capacitance, R_{in} is the input resistance of the digitizer.

Analytically, obtaining a flat frequency response means that

$$R_{in}/(R_{in} + R_p) = C_p/(C_p + C_{in} + C_c)$$

When tuned for flat response, it can be shown that

$$R_{in}(C_{in} + C_c) = C_p R_p$$

or the time constant of the probe equals the time constant of the digitizer input.

Active Probes

Active probes, such as differential and field-effect transistor (FET) probes, contain active circuitry in the probe itself to reject noise and amplify the signal. FET probes are useful for low-voltage measurements at high frequencies and differential probes are noted for their high common-mode rejection ratio (CMRR) and nongrounded reference.

Current Probes

Current probes magnetically measure AC and/or DC current flowing in a conductor instead of using a series resistance in the loop to measure current. This lack of series resistance causes very little interference in the circuit being tested.

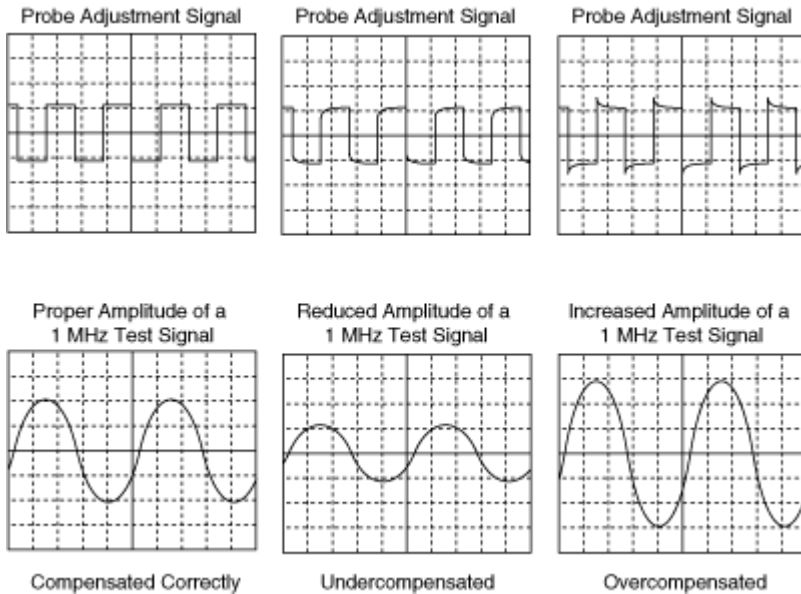
Related Topics

[Probe Compensation](#)

Probe Compensation

Adjusting the tunable probe capacitor to get a flat frequency response is called **probe compensation**. Compensating passive probes increases the accuracy of your measurements by matching the capacitance of your probe to the capacitance of a particular oscilloscope input channel.

The following figure illustrates that an undercompensated probe attenuates higher frequency signals, whereas an overcompensated probe amplifies higher frequencies.



Refer to the module or probe user documentation for information on how to compensate the probe.



Tip For the most accurate measurements:

- Compensate probes for each channel of the oscilloscope;
- Use a compensated probe only with the channel you used to compensate it; and
- Compensate your probes frequently.

Accuracy

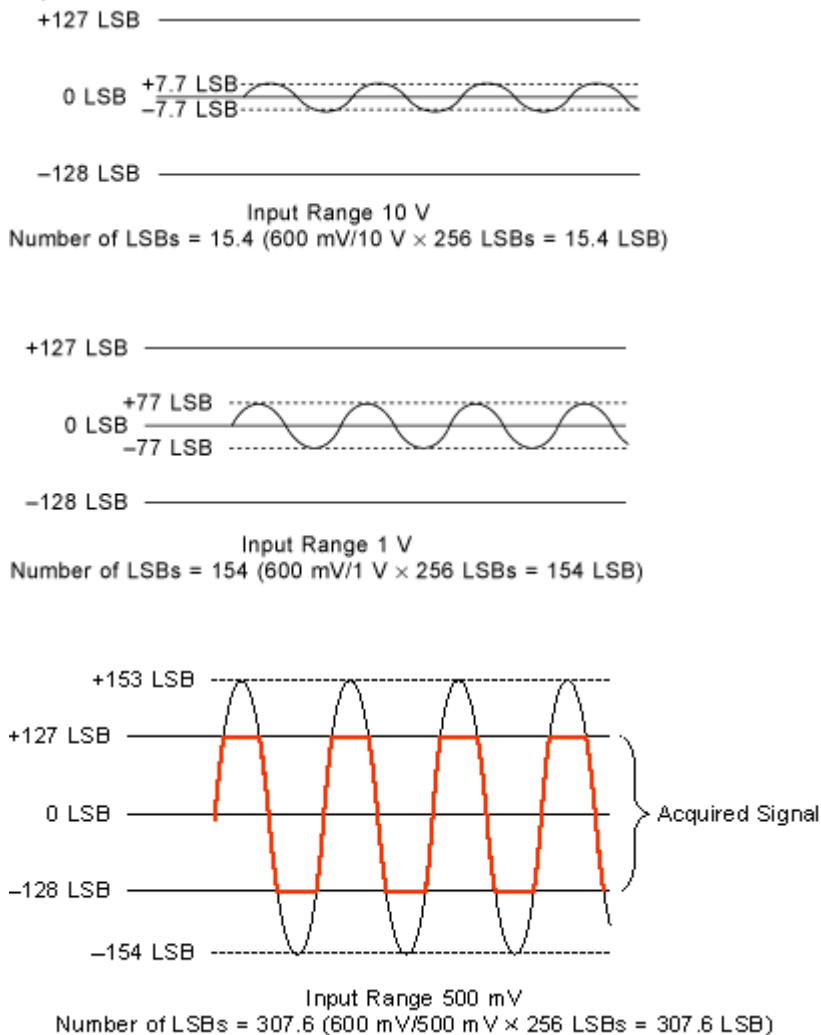
Accuracy is a measure of the ability of a device to indicate the true value of a measured signal. Accuracy is usually expressed as a percentage of the specified value, for example, $5\text{ V} \pm 1\%$.

Signal Characteristics

Knowing the characteristics of the signal under consideration helps you to choose the correct settings to maximize measurement accuracy. Such characteristics include:

- **Peak-to-peak value**—This parameter, in units of volts, reflects the maximum change in signal voltage. If V is the signal voltage at any given time, then V_{pk}

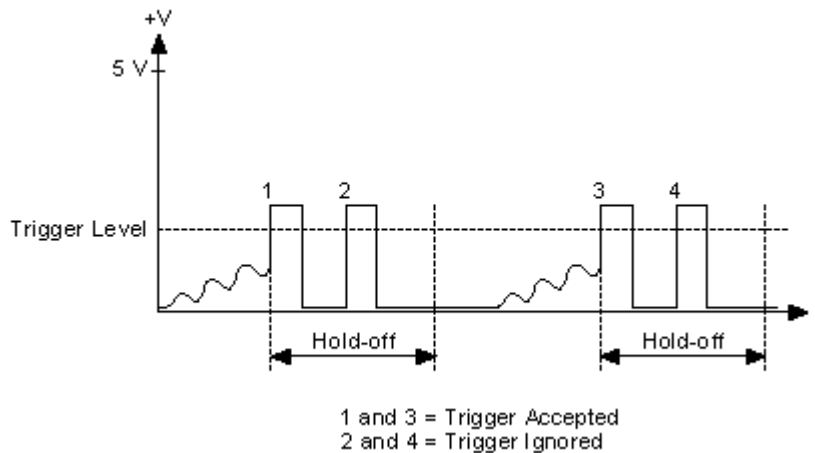
$pk = V_{max} - V_{min}$. The peak-to-peak value affects the vertical sensitivity or gain of the input amplifier. If you do not know the peak-to-peak value, start with the largest input range, and decrease it until the waveform is digitized using the maximum dynamic range without clipping the signal. Refer to the specifications for your digitizer for the maximum input range. The following figure shows how different ranges affect the resolution of a 600 mV_{pk-pk} signal.



- **Source impedance**—Most digitizers have a 1 MΩ input resistance in the passband. If the source impedance is large, the signal is attenuated at the amplifier input and the measurement is inaccurate. If the source impedance is unknown but suspected to be high, change the attenuation ratio on your probe and acquire data. In addition to the input resistance, all digitizers and

probes present some input capacitance in parallel with the resistance. This capacitance can interfere with your measurement in much the same way as the resistance does.

- **Input frequency**—If your sample rate is less than twice the highest frequency component at the input, the frequency components above half your sample rate will alias in the passband at lower frequencies, indistinguishable from other frequencies in the passband. If the highest frequency of the signal is unknown, you should start with the digitizer maximum sample rate to prevent aliasing and reduce the digitizer sample rate until the display shows either enough cycles of the waveform or the information you need. Refer to the [Nyquist Theorem](#) for more information.
- **General signal shape**—Some signals are easy to capture by ordinary triggering methods. A few iterations on the trigger level may finally render a steady display. This method works for sinusoidal, triangular, square, and saw tooth waves. Some of the more elusive waveforms, such as irregular pulse trains, runt pulses, and transients, may be more difficult to capture. The following figure shows an example of a difficult pulse-train trigger.



Ideally, the trigger event should occur at condition one, but sometimes the instrument may trigger on condition two because the signal crosses the trigger level. You can solve this problem without using complicated signal processing techniques by using [trigger holdoff](#), which lets you specify a time from the trigger event to ignore additional triggers that fall within that time. With an appropriate holdoff value, the waveform shown in the previous figure can be properly captured by discarding conditions two and four.

- **Input coupling**—On many digitizers, you can configure the input channels to be DC-coupled or AC-coupled. DC coupling allows DC and low-frequency components of a signal to pass through without attenuation. In contrast, AC coupling removes DC offsets and attenuates low-frequency components of a signal. This feature can be exploited to zoom in on AC signals with large DC offsets, such as switching noise on a 12 V power supply. Refer to the specifications for your digitizer for input limits that you must observe regardless of coupling.

Noise

Noise is any unwanted signal that appears in your sampled data. Noise comes from both external sources, such as the AC power line, motors, generators, transformers, fluorescent lights, soldering irons, CRT displays, computers, electrical storms, welders, and radio transmitters, and internal sources, such as digital clocks, microprocessors, and switched mode power supplies.

Noise can be transient in nature, have fixed frequencies such as harmonic or mixer products, or be broadband random noise.

PCI and PXI Ground Loop Noise

Some computer chassis can induce ground loop noise in measurements taken with instruments having grounded input connectors. Spurious DC and low-frequency content can result. The cause is usually power supply return currents from the motherboard that partially flow through the computer chassis to the power supply housing rather than entirely through the power supply cabling.

Follow these guidelines to mitigate the ground loop noise for PCI devices:

1. Ensure that the motherboard is tightly screwed down to the chassis so that chassis return currents will flow through the chassis under the motherboard and not in the vicinity of the PCI boards.
2. Ensure that the power supply cable is well seated on the motherboard.
3. Locate analog input cards in slots furthest away from the power supply.
4. Locate video cards and other high power PCI boards in slots furthest removed from the analog input cards.

Follow these guidelines to mitigate the ground loop noise for PXI devices:

1. Ensure that the screws on the front panel of the device are tightened down.
2. Ensure that any external equipment that is providing a signal, and is not floating with respect to ground (for example, it is not a passive sensor or a battery-operated device), is connected to the same ground as the chassis.
3. On rack mounted systems, ground the chassis to the metal rack mount to reduce ground loop currents.

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).

- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

Triggering

A trigger is an external stimulus that initiates one or more instrument functions. Trigger stimuli include digital edges, software functions, and analog levels. The trigger can be derived from attributes of the signal to be acquired, such as the level and slope of the signal.

You can use several kinds of triggering with NI high-speed digitizers, including [digital](#), [edge](#), [glitch](#), [hysteresis](#), [immediate](#), [runt](#), [software](#), [video](#), [width](#), and [window](#) triggering. Each type of triggering uses a different NI-SCOPE Configure Trigger function.

The following table describes the triggers supported by NI high-speed digitizers. The **Supported Types** column denotes which trigger types are valid for a given trigger.

Trigger Name	Supported Types	Description
Start	Digital Edge	The Start trigger transitions a device into a state where the device can respond to Sample clocks. This trigger transitions the device from a nonsampling state into a sampling state—the device starts acquiring and storing data.
Reference	Analog , Digital , Video	The Reference trigger transitions a device from the Wait for Reference Trigger sampling state into the Post Reference trigger sampling state. In the Post Reference Trigger sampling state, a counter begins counting Sample clock cycles. When the configured number of samples is acquired, the device transitions into a Done state. In other words, the arrival of this trigger establishes the reference point that separates pretrigger and posttrigger samples.
Advance	Digital Edge	The Advance trigger initiates the acquisition of the additional records in a multirecord acquisition.
Arm Reference	Digital Edge	The digitizer acquires pre-reference triggers until the Arm Reference trigger is received. For more information, refer to SMC-Based Digitizers Acquisition Engi

[ne State Diagram.](#)

Related topics:

- [Trigger Parameters](#)
- [Time Stamping](#)

Trigger Types

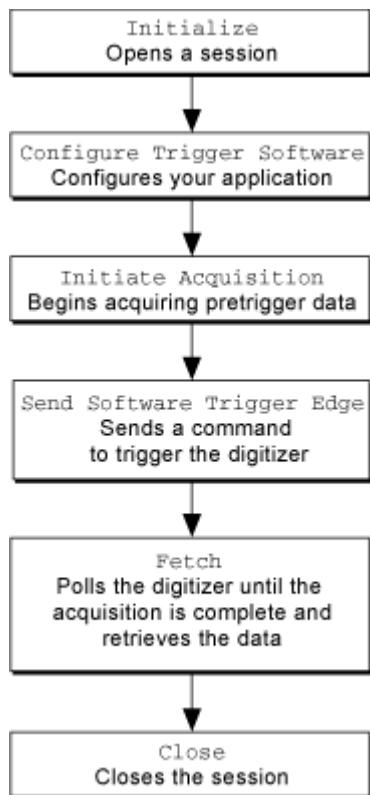
You can use several kinds of triggering with NI high-speed digitizers, including [digital](#), [edge](#), [glitch](#), [hysteresis](#), [immediate](#), [runt](#), [software](#), [video](#), [width](#), and [window](#) triggering. Each type of triggering uses a different NI-SCOPE Configure Trigger function.

Immediate Triggers

Immediate triggers occur when the digitizer triggers itself. There is no external signal that triggers the acquisition—the acquisition simply begins immediately after being configured by the driver software. Immediate triggering is the default option on NI high-speed digitizers. Therefore, you can omit using this function if you never change triggering modes.

Software Triggers

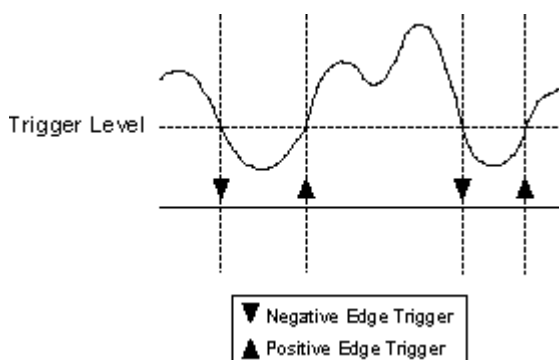
Software triggers occur when a software command starts the acquisition of posttrigger data. When you call `Initiate Acquisition` the digitizer starts acquiring pretrigger data and continues to do so until you call `Send SW Trigger`. The digitizer continues to store posttrigger samples after the software trigger, so you still need to wait for the acquisition to complete after the trigger. You can do this by setting the timeout parameter in a Fetch function to a positive value. You cannot use a Read function in conjunction with software triggering; you must use `Initiate Acquisition` followed by one of the Fetch functions. The following figure shows the programming flow using software trigger functions.



Note Not all digitizers support software triggering. Refer to [Features Supported by Device](#) to find out which digitizers support software triggering.

Edge Triggers

An edge trigger occurs when a signal crosses a trigger threshold that you specify. You can specify the slope as either positive (on the rising edge) or negative (on the falling edge) to the trigger. Edge triggering is possible on all analog trigger channels, such as 0, 1, or the external trigger channel. The following figure shows edge triggering.



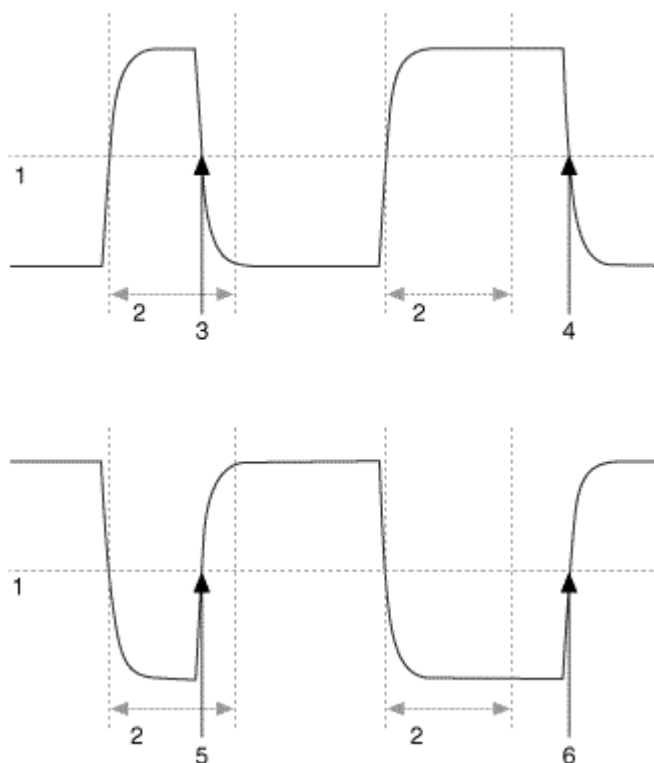
Glitch Triggers

A glitch trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a width that is either greater than or less than a duration you specify. The oscilloscope can trigger on pulses of positive, negative, or either polarity.

Use the glitch trigger to detect transient pulses within pulse trains.

You can configure the glitch parameters in NI-SCOPE using the Analog Glitch Ref Trigger instance of the [Configure Trigger \(poly\) VI](#).

The following figure illustrates glitch triggering.



Legend

1. Trigger level
2. Width
3. Positive glitch trigger (less than)
4. Positive glitch trigger (greater than)

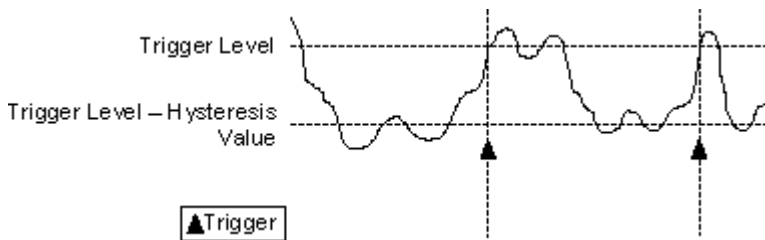
5. Negative glitch trigger (less than)
6. Negative glitch trigger (greater than)

Hysteresis Triggers

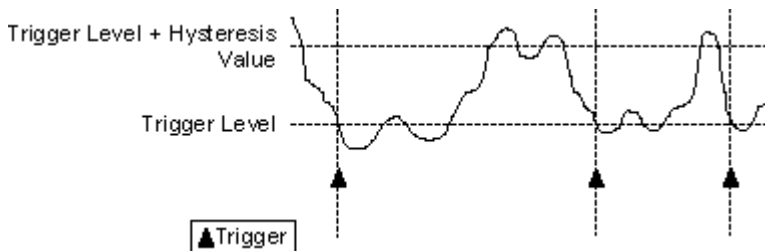
Hysteresis triggers eliminate incorrect triggers caused by noisy signals. For example, if your signal contains two rising edges of different amplitudes, you can use hysteresis triggering to trigger on one of the edges. Although NI-SCOPE uses a default amount of hysteresis for edge triggering, which is typically 2.5% of the vertical range, you can override that value by setting your own hysteresis values. The `Configure Trigger Hysteresis` function or VI allows you to choose the trigger coupling, trigger level, hysteresis value, and trigger slope.

Hysteresis triggering is possible on all analog trigger channels, such as 0, 1, or the external trigger input.

A positive slope hysteresis trigger is generated when the signal crosses below the voltage specified by the trigger level parameter minus the hysteresis value, and then crosses the trigger level.



A negative slope hysteresis trigger is generated when a signal crosses above the voltage specified by the trigger level parameter plus the hysteresis value, and then crosses the trigger level.



Runt Triggers

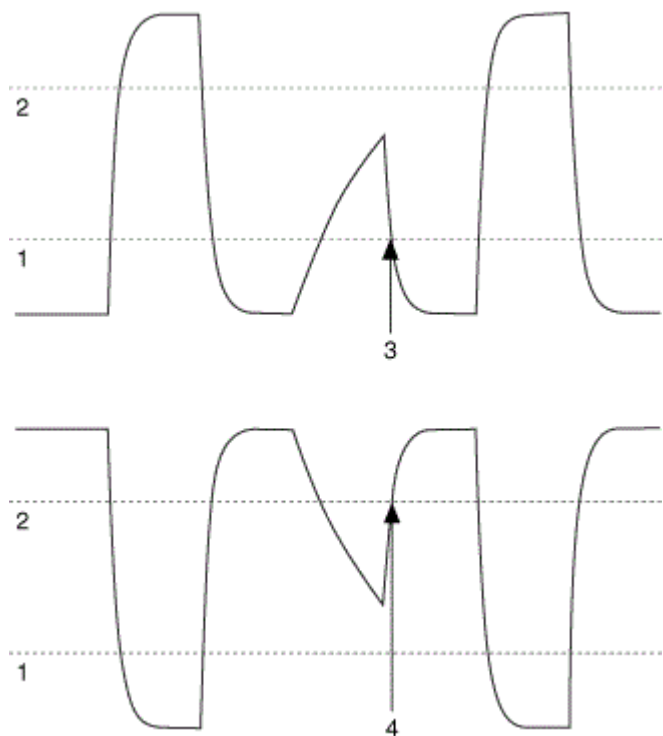
A runt trigger occurs when both the leading edge and trailing edge of a pulse cross only one of two trigger thresholds you specify and with a polarity you specify, where the polarity is relative to the threshold crossed. The oscilloscope can trigger on pulses of positive, negative, or either polarity.

You can also add time qualification to the runt trigger. A time-qualified runt trigger captures only those runt pulses that, in addition to meeting your vertical threshold criteria, have a width either within or outside a duration range you specify. By default, the runt trigger is not time qualified.

Use the runt trigger to debug digital systems by capturing pulses that could represent metastable logic states.

You can configure the runt trigger parameters in NI-SCOPE using the Analog Runt Ref Trigger instance of the [Configure Trigger \(poly\) VI](#). Use NI-SCOPE properties to time-qualify the runt trigger.

The following figure illustrates runt triggering.



Legend

1. Runt low threshold
2. Runt high threshold
3. Positive runt trigger
4. Negative runt trigger

Video Triggers

A video trigger occurs when the digitizer finds a valid video signal synchronization. Refer to [NI 5114/5122/5124/5142 Video Triggering](#) for more information.

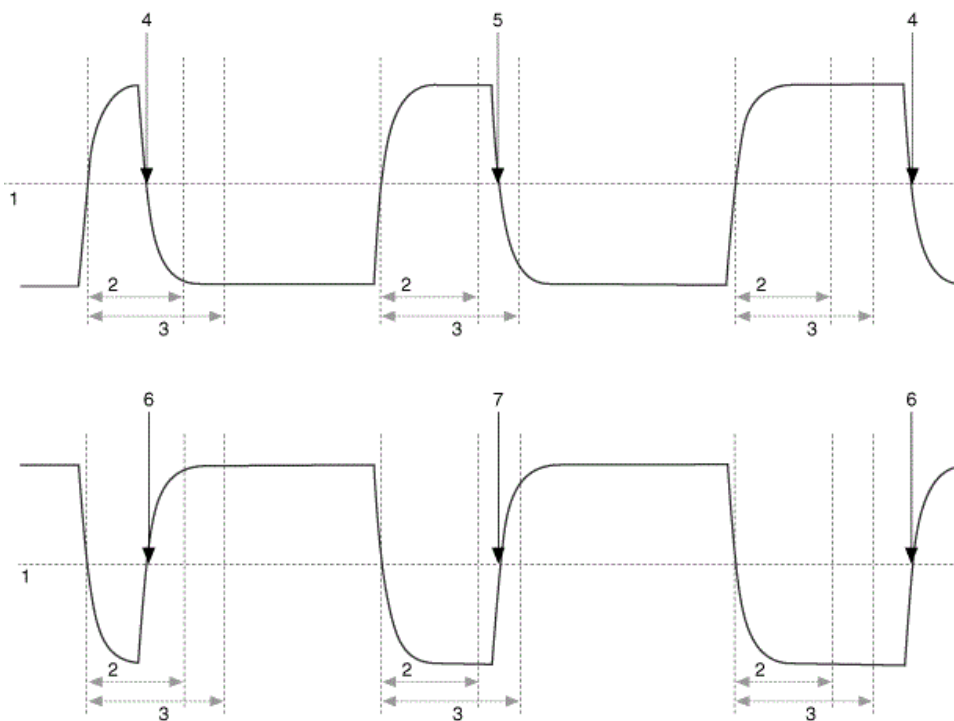
Width Triggers

A width trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a duration that is either within or outside a duration range you specify. The oscilloscope can trigger on pulses of positive, negative, or either polarity.

Use the width trigger to detect transient pulses with a duration that falls within or outside of a known range.

You can configure the width trigger parameters in NI-SCOPE using the Analog Width Ref Trigger instance of the [Configure Trigger \(poly\) VI](#).

The following figure illustrates width triggering.



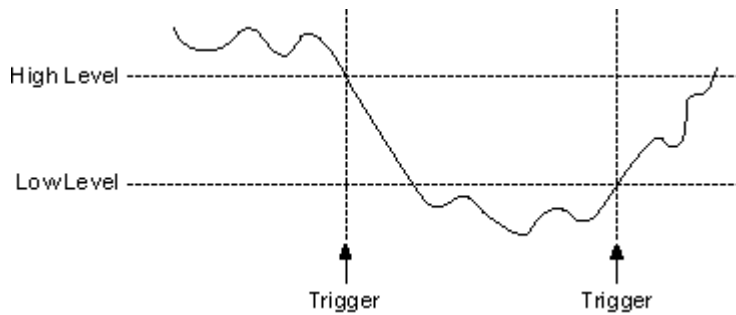
Legend

1. Trigger level
2. Width low threshold
3. Width high threshold
4. Positive width trigger (outside)
5. Positive width trigger (within)
6. Negative width trigger (outside)
7. Negative width trigger (within)

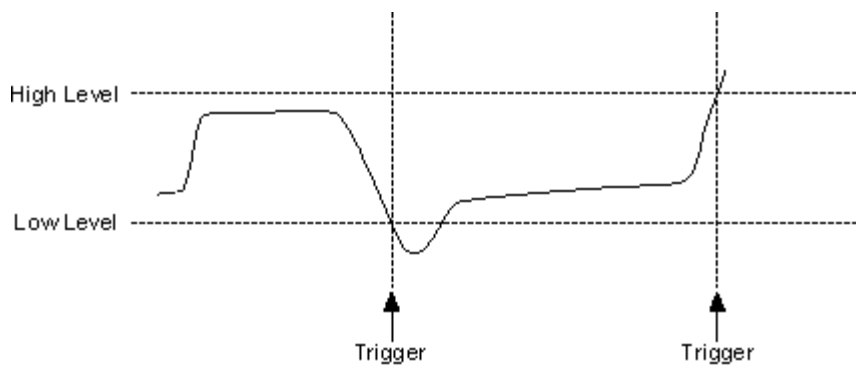
Window Triggers

A window trigger occurs when a signal either enters or leaves a window you specify with the window mode parameters in the `Configure Trigger Window` function. Window triggering is possible on all analog trigger channels, such as 0, 1, or the external trigger input.

The following figure shows an entering window trigger.



The following figure shows a leaving window trigger.



Digital Triggers

A digital trigger occurs on either a rising edge or falling edge of a digital signal. Digital triggering is possible on the [RTSI](#) lines, [PFI](#) lines, and the [PXI Star Trigger](#) line.

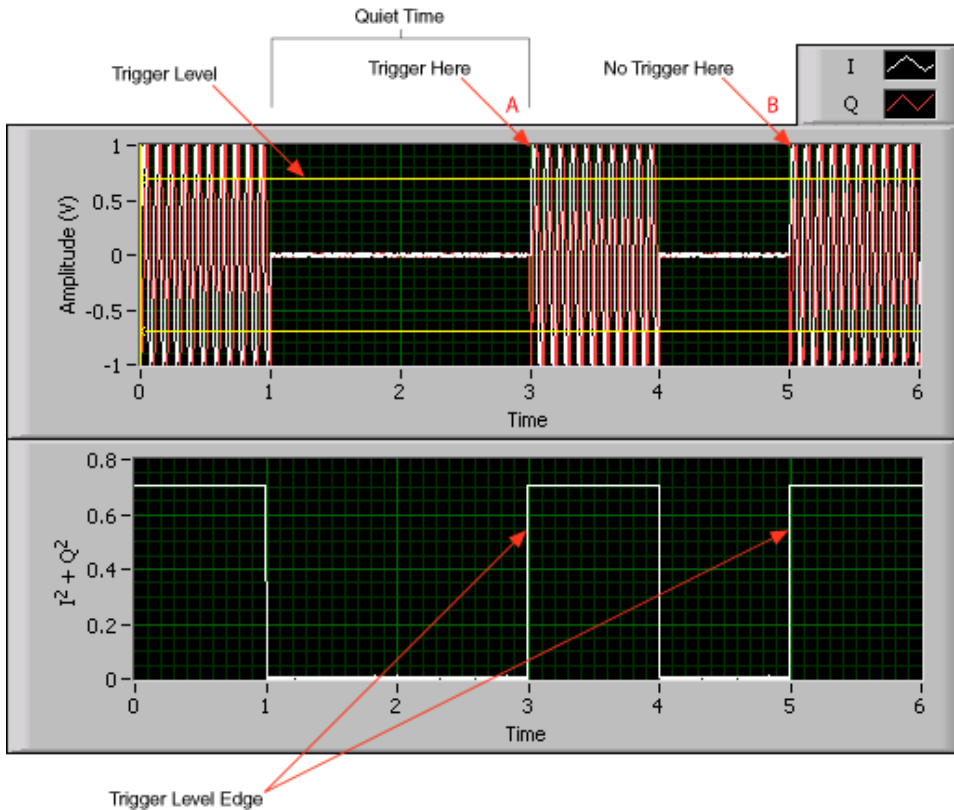
Burst Triggers



Note Burst triggering is available only on high-speed digitizers that support onboard signal processing (OSP).

To trigger on data bursts, you can configure the digitizer to wait until it detects a quiet period before arming the trigger. Use the [Min Quiet Time](#) property or the [NISC OPE_ATTR_REF_TRIGGER_MINIMUM_QUIET_TIME](#) attribute to set the length of this quiet period.

The following figure shows how burst triggers occur after a configured minimum quiet time of 1.5 seconds. The trigger occurs at point A because the minimum quiet time of 1.5 seconds has been exceeded—in this case, 2 seconds have elapsed since a signal above the trigger level was detected. A trigger does not occur at point B because only 1 second of quiet time has elapsed.



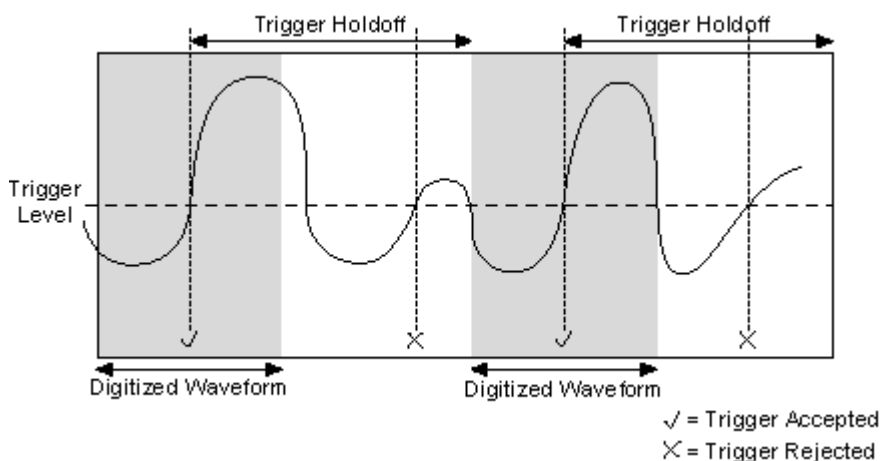
Trigger Parameters


Common NI-SCOPE trigger parameters include [trigger holdoff](#), [trigger delay](#), and [trigger coupling](#).

Trigger Holdoff

Trigger holdoff is an adjustable period of time during which the digitizer cannot trigger. NI high-speed digitizers use trigger holdoff to guarantee a minimum time between two Reference Triggers. This feature is useful if incoming triggers are a known time apart and only a relatively small window of samples adjacent to the trigger are necessary. In this case, setting the holdoff allows you to efficiently allocate the onboard memory, and prevents the device from triggering on sections of the input waveform that are not of interest. Each device uses an onboard counter that is programmed by software to implement this functionality. While the holdoff counter is decrementing, incoming Reference Triggers are ignored. The device arms itself for the next Reference Trigger when the holdoff counter reaches zero. Each

device that supports trigger holdoff has a minimum holdoff value, which is listed in the specifications document for each device.



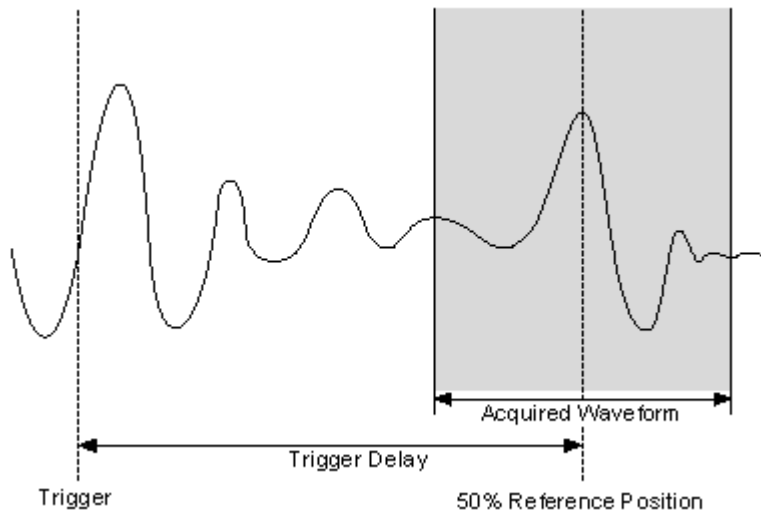
 Note Trigger holdoff is not supported by all digitizers. Some digitizers allow only trigger holdoff or trigger delay at any given time. Refer to [Features Supported by Device](#) to find out which digitizers support trigger holdoff.


The details of how holdoff is implemented on each device varies. Refer to the device-specific help sections for more information on your particular digitizer.

Trigger Delay

Trigger delay is the amount of time that elapses between when a trigger occurs and when the digitizer actually triggers.

In NI-SCOPE, the trigger delay parameter specifies this time period, (in seconds). For example, setting a 1 second trigger delay means the digitizer continues to acquire pretrigger samples for 1 second after it is triggered. The posttrigger samples occur after the delayed trigger.




 **Note** Trigger delay is not supported by all digitizers. Some digitizers allow only trigger holdoff or trigger delay at any given time. Refer to [Features Supported by Device](#) to find out if your digitizer supports trigger delay.

Related topics:

- [Trigger Parameters](#)

Trigger Coupling

The trigger coupling parameter can be either AC, DC, HF Reject, LF Reject, or AC Plus HF Reject. This parameter works much like the vertical coupling parameter. When AC is selected, NI-SCOPE rejects the DC component of the trigger signal. When triggering on channel 0 or channel 1 of the digitizer, the input signal is immediately coupled based on the vertical coupling parameter. Therefore, if the vertical coupling parameter is set to AC coupling, triggering occurs on the AC components of the signal because the DC component has already been removed.

 **Note** For the PXIe/PXI/PCI-5122, trigger coupling applies only to the external trigger channel. When using the analog input channels (channels 0 and 1) as the trigger source, the trigger coupling is the same as the [input coupling](#) and is specified by calling `Configure Vertical`.

Sample Clock

The sample clock is sent to the ADC of each channel and to the acquisition engine to control the sampling rate of a digitizer. The sample clock sources are as follows.

Internal Sample Clock

High-speed digitizers and oscilloscopes have an onboard (internal) clock that controls the sampling rate as well as other timing functionality of the device. In most cases, the onboard oscillator is a Voltage Controlled Crystal Oscillator (VCXO). Typically, an onboard DAC (digital to analog converter) is used to calibrate the VCXO to exactly the desired clock rate. This DAC can also be used to adjust the frequency of the VCXO to phase lock it to a [reference clock](#). The maximum sampling rate of a device is usually determined by the speed of the onboard clock. However, other sampling rates can be achieved by two methods; decimation of the data or dividing down the onboard clock.

Decimation Method

In the decimation method, the ADC samples at the rate of the onboard clock and then sends its digital data to a decimator that essentially discards samples at a specific interval to achieve slower effective sampling rates. The valid sampling rates are always an integer divisor of the onboard clock. For example, if the onboard clock is 100 MHz but you want to sample at 25 MS/s, you must use decimation. The decimation method would discard all data except for every fourth datapoint to achieve exactly 1/4 of the maximum sample rate (or onboard clock rate).

Divide Down Clock Method

Some oscilloscopes use the divide down clock method. In this method the onboard clock is sent through a series of clock dividers, and then that clock is sent to the ADC. Typically, the decimation method is preferred over the divide down clock method.



Note NI high-speed digitizers support the decimation method but do not support the divide down clock method.

External Sample Clock

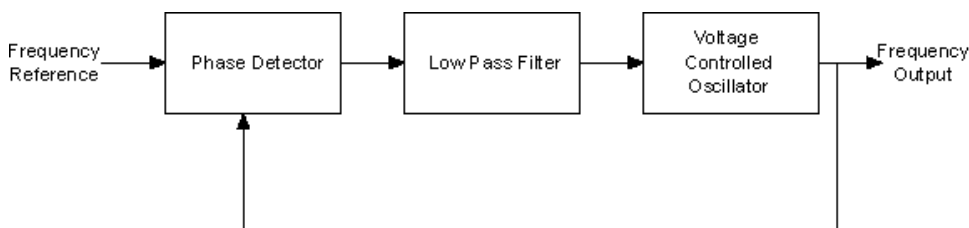
Some digitizers and oscilloscopes can accept an external sample clock. This external sample clock is used to replace the onboard clock (the VCXO) for synchronization or to achieve a sampling rate that cannot be specified by using the onboard clock. Some devices can also decimate the external sample clock to achieve a sampling rate that is an integer divisor of the external sample clock. For example, if the external sample clock is 70 MHz, you could decimate the clock by a factor of 2 and achieve a 35 MS/s sampling rate.



Note Not all devices support the option for an external sample clock and/or decimation of that clock. Refer to [Features Supported by Device](#) for more information. If your device does not support external sample clocks, refer to [reference clock](#) for another method of synchronization.

Reference Clock/Phase-Lock Loop

The reference clock is used in the phase-locked loop (PLL) circuit of a digitizer to synchronize the internal oscillator to the reference clock. The frequency stability of the internal oscillator matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The most common frequency for a reference clock is 10 MHz because a clock of that frequency can generally be shared over a cable without much attenuation or loss. However, the frequency range of reference clocks can vary anywhere from 1 MHz to 20 MHz. Refer to the [Devices](#) section of this help file to determine what your device supports. The following figure shows a block diagram of a basic PLL.



The operation of this circuit is typical of all PLLs. A PLL is a feedback control system that controls the frequency and phase of a voltage controlled oscillator (VCO). The input signal is applied to a phase detector. The output of the VCO connects to the other input. As shown in the previous diagram, the frequencies of both signals are the same. The output of the phase detector develops a voltage proportional to the

phase difference between the two input signals. The lowpass filter receives this signal from the phase detector. The lowpass filter determines the dynamic characteristics of the PLL. The filtered signal controls the VCO.

Time Stamping

NI digitizers use a free-running timing clock and/or a [time-to-digital converter \(TDC\)](#) to accurately record the time of an event. For example, the NI 5114/5122/5124/5142/5922 each have a 64-bit counter. NI-SCOPE creates accurate time records by aligning the NI-TIO clock to the sample clock on your device. This time record is a **time stamp**. With time stamps, you can correlate multiple records or even multiple acquisitions. For example, you can use time stamps to determine the amount of time between acquisitions.

How Time Stamps Work

NI-SCOPE returns time stamps with every Fetch function, if the digitizer supports them, in the waveform info structure. This structure contains both absolute initial x and relative initial x parameters. The absolute initial x parameter is the time of the first point in the waveform in units of seconds, and it has a resolution of the maximum real-time sampling period of the digitizer (such as 10 ns for a 100 MS/s digitizer). This time is valid between records and entire acquisitions, provided the clock is not reset. The relative initial x parameter, which is supported by all digitizers, is the time between the first point in your waveform and the trigger. Therefore, the trigger always occurs at time equals zero. It includes the interpolated TDC. This conversion measures when the trigger occurs between two samples. Keep in mind that the relative initial x value is not valid until a trigger occurs when using continuous acquisition. To compute an absolute trigger time that includes the TDC value, use the following formula:

$$\text{absolute trigger time} = \text{absolute initial x} - \text{relative initial x}$$

The time it takes for the time stamp counter to roll over depends on the number of bits in the time stamp counter and the clock frequency. For instance, the rollover

would occur for an NI 5114/5122/5124/5142/5922 if it runs for 5849 years. If the rollover occurs, NI-SCOPE returns a warning.

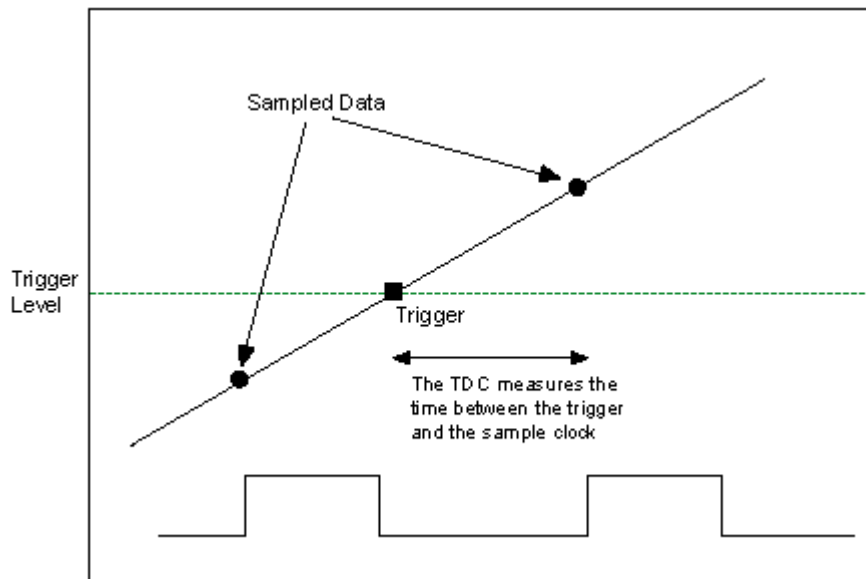
To reset the counter on the NI 5114/5122/5124/5142/5922, call `niScope Reset`.

Time Stamps Example

The `Timestamps` example uses time stamps to compute a histogram of the time between triggers. You can use this example to calculate the minimum time required between records. You can find the `Timestamps` example by using the shortcut `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

TDC

A time-to-digital converter (TDC) is a circuit that is used to measure the time between two events. For digitizers, the TDC measures the time elapsed from when a trigger is received by the digitizer to the time the first sample is taken.



Sampling Methods

The sampling method you use determines how the digitizer collects sample points. Depending on whether the signal is changing slowly or quickly, you can choose one of the following methods to collect samples:

- [Real-Time Sampling](#)
- [Random Interleaved Sampling \(RIS\)](#)
- [Time Interleaved Sampling \(TIS\)](#)

Real-Time Sampling

Real-time sampling is the default sampling method used by NI high-speed digitizers. In real-time sampling, the digitizer gathers all the samples for a waveform in one acquisition with one trigger event.

Related Topics

[Random-Interleaved Sampling](#)

Equivalent-Time Sampling and Random Interleaved Sampling

Equivalent-time sampling is a sampling method in which a picture of a waveform is created over time by using a series of samples taken from repetitive waveforms.

Random Interleaved Sampling (RIS) is a form of equivalent-time sampling that increases apparent sample rates of repetitive signals by combining several triggered waveforms. Because the trigger time occurs randomly between two samples, the digitizer samples different points in the waveform on consecutive acquisitions. By combining these waveforms, you can achieve RIS sample rates that are up to 25 times higher than the ADC sample rate on the digitizer.



Note NI-SCOPE does not support using RIS with multi-instrument NI-SCOPE sessions.

How RIS Works

Each waveform trigger occurs at some time randomly distributed between two samples. The digitizer can measure the time from the trigger to the next sample, commonly called the time-to-digital conversion (TDC), extremely accurately—with hundreds of times more resolution than the sample period of the digitizer. With these TDC measurements, NI-SCOPE can combine multiple waveforms, aligned at the trigger time, to simulate a faster sampling rate for repetitive signals.

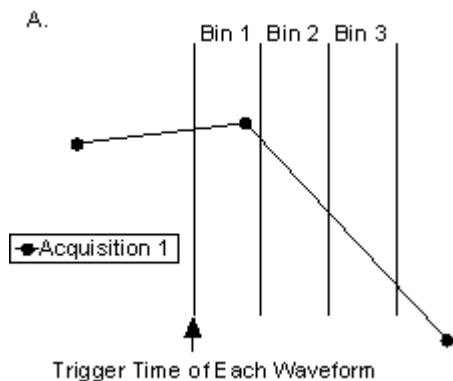
How Oversampling Factors Increase Effective Sample Rates

When a digitizer is in RIS mode, the legal sample rates become multiples of the maximum real-time sampling rate. These multiples are the oversampling factor. For example, for a digitizer that has a 100 MS/s maximum real-time sampling rate, the RIS sampling rates are the oversampling factor times 100 MS/s, where the oversampling factor is two, three, four, and so on. If you specify 300 MS/s for your sampling rate, the oversampling factor is three.

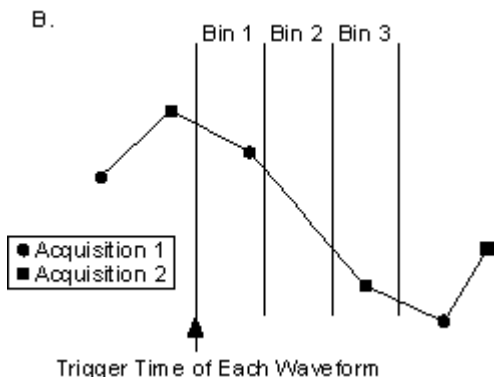
Each TDC value is between zero and the sampling period, so a 100 MS/s digitizer has TDC values between 0 and 10 ns. This time span is divided into a number of bins equal to the oversampling factor. For example, an oversampling factor of three means there is one TDC bin from 0 to 3.33 ns, another from 3.33 to 6.67 ns, and another from 6.67 to 10 ns. To reconstruct a waveform, each bin must contain at least one TDC value.

The following four figures show an example RIS acquisition with an oversampling factor of three. Each of the figures shows the aligned trigger time of the waveforms followed by the three TDC bins, where the combined width of the three bins is the real-time sample period of the digitizer.

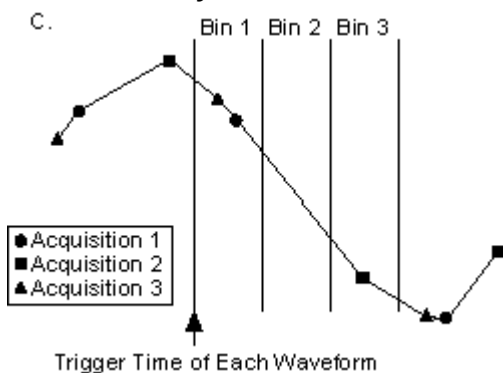
1. The first waveform has a TDC value that falls in bin number 1.



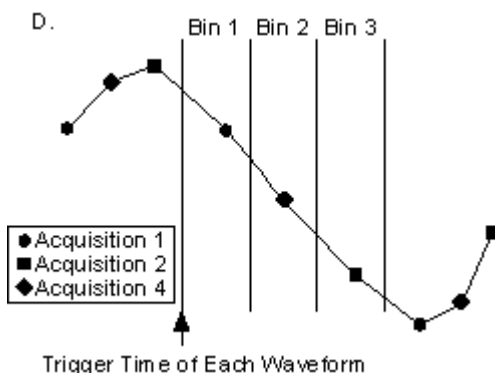
2. The succeeding waveform falls in bin number 3.



3. Because of the randomness of the TDC value, another sampled waveform falls in bin 1. NI-SCOPE does not fetch this waveform from the digitizer because the bin is already filled.



4. The next waveform falls in bin 2.

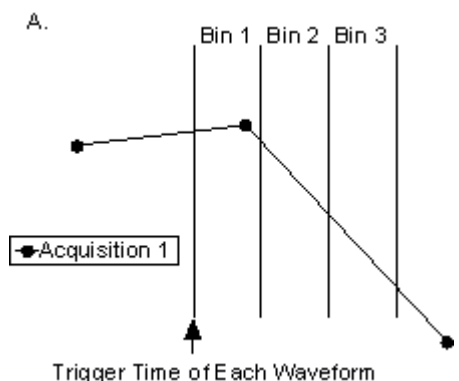


Now all bins are full, so NI-SCOPE returns the three waveforms sampled at the maximum real-time rate as one evenly-sampled waveform with a sampling rate three times greater.

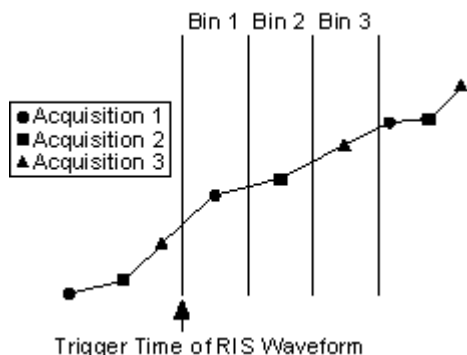
Averaging to Minimize Noise

At a minimum to complete an [RIS](#) acquisition, the number of TDC values must be equal to the oversampling factor. However, this does not guarantee a cleanly reconstructed signal. The following figures show three acquisitions of a perfectly linear edge.

The following figure shows that the sample in bin number 1 is extremely close to its right-hand bin edge. When NI-SCOPE returns the RIS waveform, this sample is coerced to the middle of the bin. In similar fashion, each sample is coerced to the center of its bin.

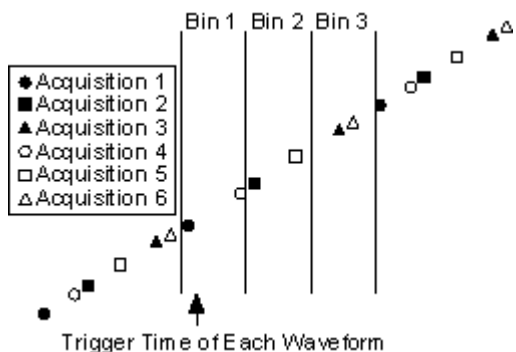


The following figure shows how NI-SCOPE returns the RIS waveform, with evenly-spaced samples.

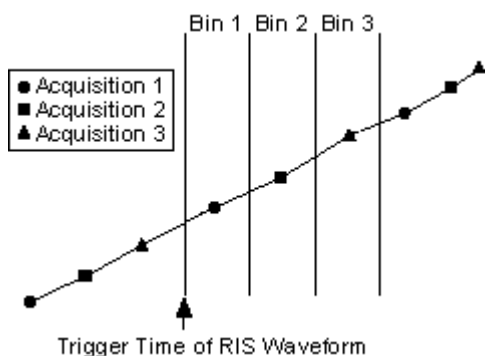


Note You can avoid noise in RIS waveforms by averaging multiple waveforms in each bin during the RIS algorithm. The number of averages in each bin can be specified with the RIS Num Averages attribute of NI-SCOPE.

The following figure shows the acquisition using two averages. Notice that twice as many waveforms are required now.



The following figure shows the returned waveform from the averaged RIS acquisition.




The higher the number of averages, the smaller the effect of coercing the time of samples. Therefore, it is generally important to use some amount of averaging with RIS. Increasing the amount of averaging increases the minimum number of waveforms necessary to reconstruct the RIS waveform and therefore, the time it takes to complete an acquisition.

Randomness in RIS Acquisitions


When using RIS, remember that the TDC value is completely random—the hardware does not adjust anything to ensure an even distribution of TDC values. If the TDC is the same from two acquisitions, the digitizer has effectively sampled the same places on the repetitive waveform, so no information has been gained. Therefore, for RIS to work, you must have a suitable random distribution of TDC values, which requires a nondeterministic number of waveforms. While it can take a long time to receive a suitable set of waveforms with different TDC values, in practice you rarely need to wait long. Furthermore, NI-SCOPE employs multirecord acquisitions and

continuous acquisition when possible to optimize reconstructing the waveform while the acquisition is in progress.

 **Note** Sometimes an RIS acquisition times out and does not complete. If increasing the timeout value does not solve the problem, you may need to [self-calibrate](#) the digitizer. This timeout can also occur if the digitizer is frequency locked to the same reference as the frequency source.

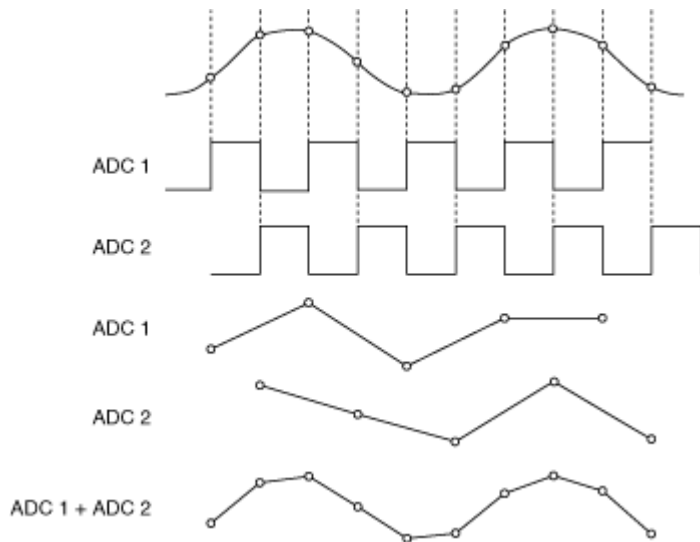
Time Interleaved Sampling

Time Interleaved Sampling (TIS) is a technique to increase the real-time sample rate of a digitizer. TIS uses multiple analog-to-digital converters (ADCs) to sample the same input waveform, but at different relative phases. The hardware then interleaves these samples to create the waveform as if only one ADC were sampling the waveform at a higher sample rate.

 **Note** To use time interleaved sampling for the NI 5152, 5153, and 5154, set the [Enable TIS](#) property or the `NISCOPE_ATTR_ENABLE_TIME_INTERLEAVED_SAMPLING` attribute to TRUE. For all other devices, TIS is enabled automatically by setting the sample rate. The Enable TIS property is ignored.

How TIS Works

When you enable TIS on a channel, the digitizer samples the input signal of that channel using more than one ADC. The clocks of these ADCs are shifted to a phase that reflects the time when an ADC of a higher sample rate would acquire a point. An example is shown in the following figure.



To obtain a 2 G/s TIS rate using two ADCs, each operated at 1 G/s, the 1 GHz clock of one ADC is shifted 180 degrees with respect to the 1 GHz clock of the other ADC.

The data returned is in order and is acquired in real time. Therefore, the input signal is not required to be repetitive.

How TIS Affects Measurements

Because multiple ADCs are used in TIS, it is possible for mismatch between ADCs and phase offset between clocks to cause small changes in the spectral performance of the digitizer. NI digitizers calibrate most of these changes out, but some small residual changes may persist. It may be important, particularly in frequency domain applications, to understand the following effects.

- **Offset Mismatch**—If the ADCs are not perfectly matched in DC offset, a small spur can appear at DC and at the Nyquist frequency (Sample Rate/2).
- **Gain Mismatch**—If the ADCs are not perfectly matched in DC gain, spurs occur at image frequencies of input signal frequencies. The size of the image spurs are proportional to the amount of gain mismatch.
- **Phase Offset**—If the ADCs are not sampling at exactly the correct phase for the interleaving factor, images of the frequency content similar to those found with gain mismatch will occur.

Frequency Domain Fundamentals

Expand this book to view the frequency domain topics, or click one of the following links:

[SFDR](#)

[THD](#)

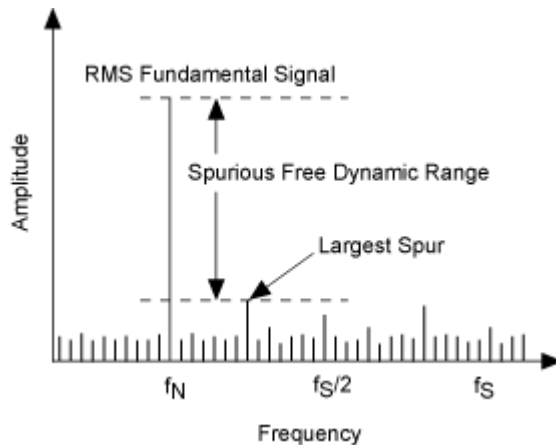
[SINAD](#)

[ENOB](#)

Spurious Free Dynamic Range

Spurious free dynamic range (SFDR) is the usable dynamic range before spurious noise interferes with or distorts the fundamental signal. The amplitude of the fundamental signal is usually -1 dBFS. SFDR is the measure of the ratio in amplitude between the fundamental signal and the largest harmonically or non-harmonically related spur from DC to the full Nyquist bandwidth (half the sampling rate). A spur is any frequency bin on a spectrum analyzer, or from a Fourier transform, of the analog signal. SFDR is expressed in dBc.

The following figure illustrates how SFDR is measured.



THD

The total harmonic distortion (THD) of a signal is the ratio of the sum of the powers of the first five harmonics above the measured fundamental frequency to the power

of the fundamental frequency. THD is usually expressed in dB or dBc. Measurements for calculating the THD are made at the output of a device under specified conditions.

SINAD

Signal-to-noise-and-distortion (SINAD) is the ratio of the rms signal amplitude (set to 1 dB below Full Scale) to the rms sum of all other spectral components, including the harmonics but excluding DC. SINAD is usually expressed in dB.

ENOB

Effective number of bits (ENOB) is another way of specifying signal-to-noise and distortion ratio (SINAD). ENOB is calculated using the following formula.

$$\text{ENOB} = \frac{(\text{SINAD} - 1.76) + 20 \log\left(\frac{V/2}{A}\right)}{6.02}$$

where **A** is the input amplitude and **V** is the full scale range of the ADC.

The ENOB indicates that the ADC is equivalent to an ideal ADC of this number of bits.

Video Fundamentals

Expand this topic for information about the fundamentals of video signals.

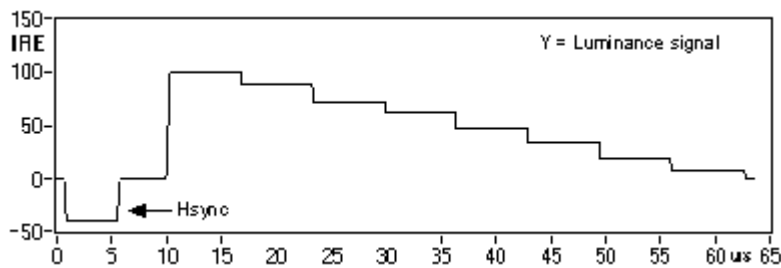
Composite and Component Video Signals

Composite Video Signals

A composite video signal is a signal in which all the components needed to generate a video signal are embedded in a single signal. The three main components that together form a composite signal are as follows:

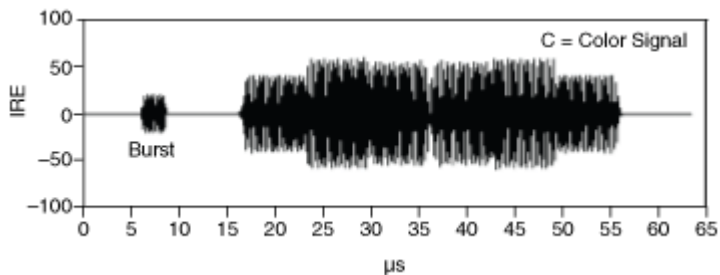
- The luma signal (or luminance)—Contains the intensity (brightness or darkness) information of the video image
- The chroma signal—Contains the color information of the video image
- The synchronization signal—Controls the scanning of the signal on a display such as the TV screen

The monochrome composite signal is built of two components: luma (or luminance) and synchronization. This signal, which is usually called the **Y** signal, is shown in the following figure.



Monochrome Composite Video Signal (Luma Steps from White to Black)

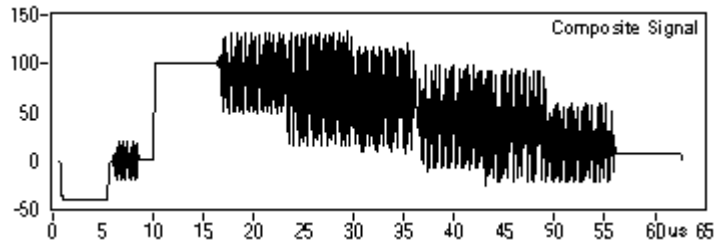
The chroma signal by itself, which is usually called the **C** signal, is shown in the following figure.



Color Information Signal for a Color Bar Line (Including the Color Burst)

The composite color video signal, often called the Color Video, Blank, and Sync (CVBS) signal, is the sum of **Y** and **C**.

$$\mathbf{CVBS = Y + C}$$



Color Composite Video Signal for a Color Bar Line

The two components **Y** and **C** can also be distributed separately as two independent signals. These two signals together are called either Y/C or S-video.

Y/C = S-video

Component Video Signals

A component video signal is a signal in which all the color components needed to generate a video signal are transmitted separately. Most HDTV formats are component (Y-Pb-Pr).

Video Format Naming

The most common standard video formats (NTSC, PAL, and SECAM) are known by acronyms. Sometimes extra letters are appended to the front of the names to indicate the refresh rate. For example, M-PAL and M-NTSC have a refresh rate of 59.94 fields per second.

Standard Definition (SDTV) Formats

The definition of standard composite video signal formats differs from country to country.

- M-NTSC—Mainly used in the USA and Japan.
- B/G-PAL—Used in most of Europe, parts of Africa, and some other countries such as Australia.
- SECAM—Used in France, the former Soviet Union, and parts of the Middle East.
- M-PAL—Used in Brazil.

EDTV and HDTV Video Formats

Other video signal formats such as HDTV (high-definition television) and EDTV (enhanced-definition television) are named using characteristics of the video format. For example, an HDTV video format with 1,080 active horizontal lines using an interlaced scheme with a refresh rate of 30 fields per second is designated as 1080i/30 fps. If the video signal format is changed to use a progressive scheme, the name would change to 1080p/30 Fps.



Note Fps = Frames per second, used in progressive formats; fps = fields per second, used in interlaced formats.

EDTV Formats

- 480i/59.94 fps
- 480i/60 fps
- 480p/59.94 Fps
- 480p/60 Fps
- 576i/50 fps
- 576p/50 Fps

HDTV Signal Formats

- 720p/30 Fps (not supported by NI-SCOPE)
- 720p/50 Fps
- 720p/59.94 Fps
- 720p/60 Fps
- 1080i/50 fps
- 1080i/59.94 fps
- 1080i/60 fps
- 1080p/24 Fps

Scanning Speed

The scanning speed is the number of video frames generated per second. The scanning speed and the number of lines per frame depend on the video format.

NTSC specifies a scanning speed of approximately 30 frames per second and 525 lines per frame while PAL and SECAM specify a rate of 25 frames per second and 625 lines per frame.

Color Coding

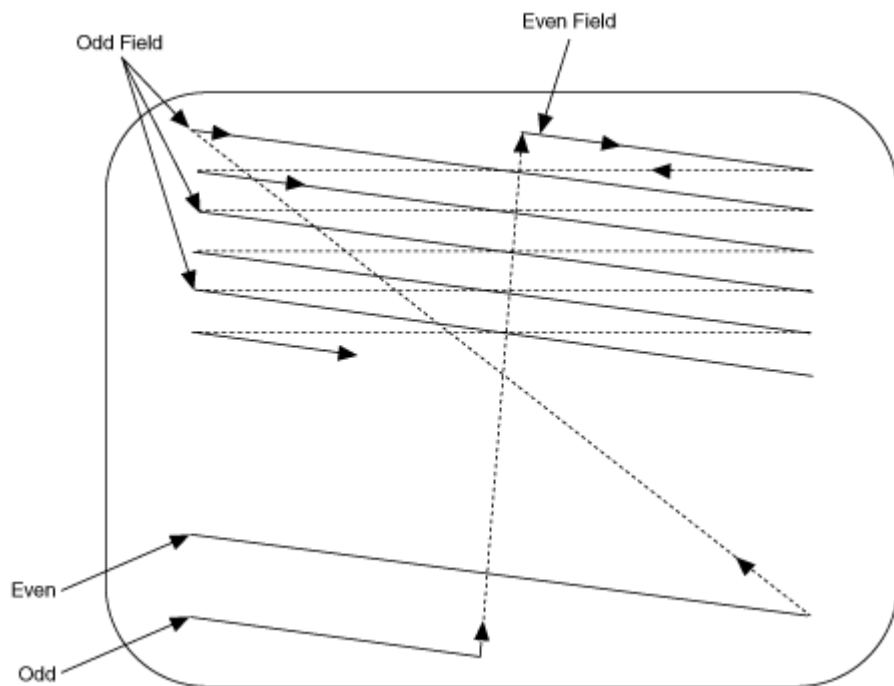
The color information signal C, together with Y, forms the composite video signal is a modulated signal. The modulation type depends on the video format.

For all PAL and NTSC formats, the coding is based on the Quadrature Amplitude Modulation (QAM) concept, where 2-color components are amplitude modulated in quadrature and then combined. The modulation must be decoded, so to keep track of the absolute phase needed to decode the color information, a reference signal, called the color burst, is inserted at the beginning of each line, right after the horizontal synchronization pulse. The insertion signal is shown in the [Complete NTSC Frame Scan](#) image.

For the SECAM format, the 2-color components are frequency modulated using two different subcarrier frequencies and are sequentially distributed on alternated video lines. SECAM does not need a color burst signal.

Interlaced Scanning

All composite video systems display the video image on a TV screen using an interlaced scanning technique. The following figure shows the interlaced scanning concept.



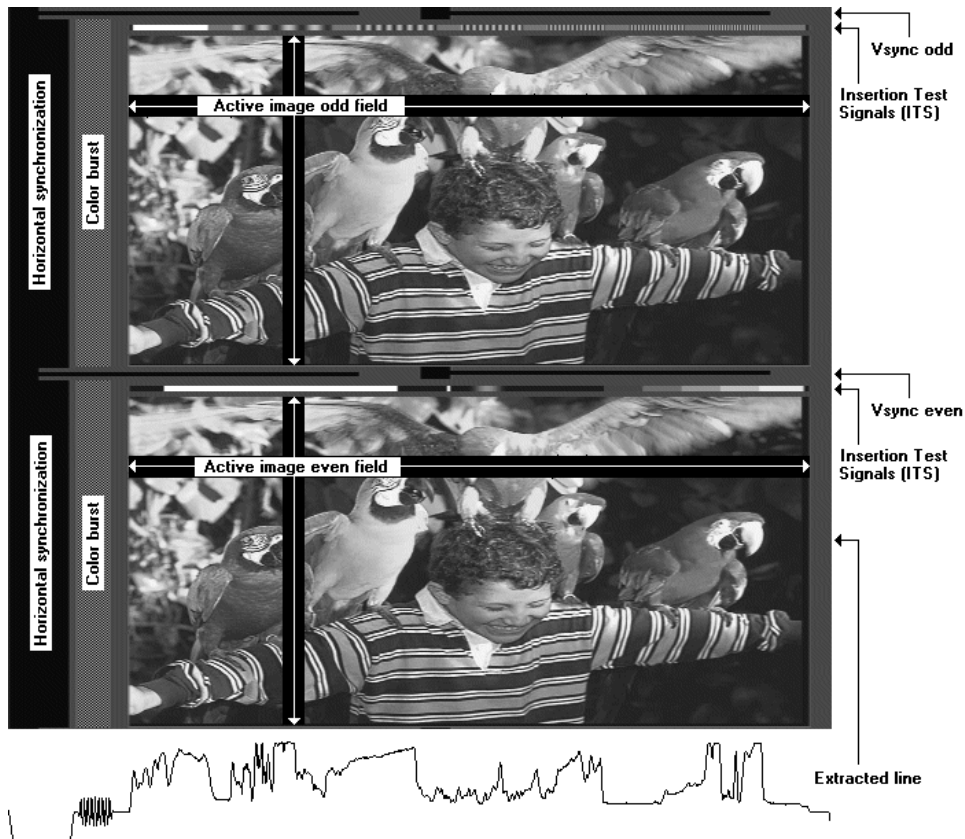
Interlaced Scanning Concept

The analog video signal includes synchronization pulses that control the scanning line-by-line from left to right and field-by-field from top to bottom. The pulses that control the line-by-line scanning are called the horizontal synchronization pulses (Hsync). The pulses that control the vertical scanning are called the vertical synchronization pulses (Vsync).

Two interlaced fields compose a complete frame. The first field, called the odd field, scans the odd lines of the video image. The second field, called the even field, scans the even lines of the video image. The process repeats for every frame.

Complete NTSC Frame Scan

The following figure shows the result of scanning all 525 lines that compose a complete NTSC frame.



Gray Scale Image and Extracted Line Profile

The [Complete NSTC Frame Scan](#) image simulates the video display that would appear on a television screen if the following conditions were true:

- The television could show the entire line instead of just the [active image](#) part.
- The television was not interlacing the two fields to form a complete image frame, but instead was displaying a progressive scanning, line by line, of the entire frame.

The scanning starts (line-by-line from top to bottom) with a number of lines that represent the vertical synchronization pattern for the odd field. Immediately after the vertical synchronization pattern for the odd field, optional insertion test signals (ITS) are inserted. Finally, the actual odd field active image displays.

The process repeats for the even field, forming the complete frame.

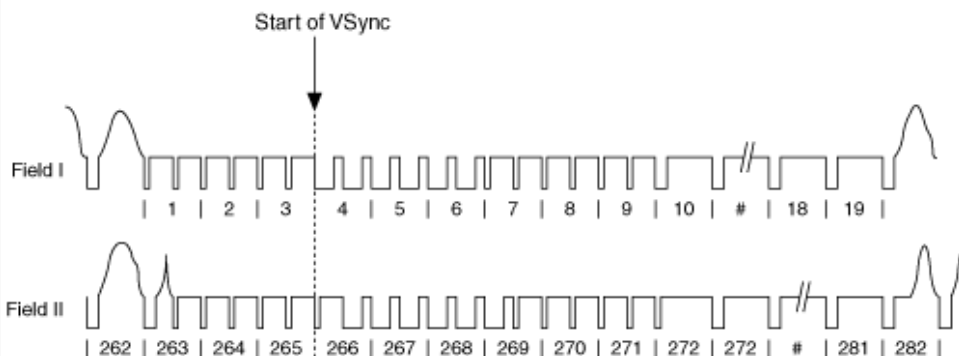


Note Most lines start with a horizontal synchronization pulse followed by the color burst pattern. Then the active image (or the ITS) displays as an intensity change, where a higher signal level corresponds to brighter intensity.

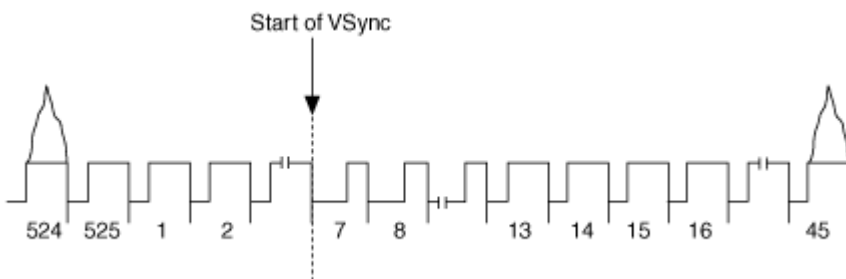
The extracted line profile example at the bottom of the figure shows an actual video signal line extracted from the even field. Refer to [Video Levels](#) for more information.

Horizontal synchronization pulses are basically simple negative pulses, which are pulses going below the level of the luminance signal. However, the vertical synchronization signals are composed of pulse trains distributed on several lines, and the pulse trains are different for odd and even fields. The following figures show the vertical synchronization patterns for both fields and for several main video formats.

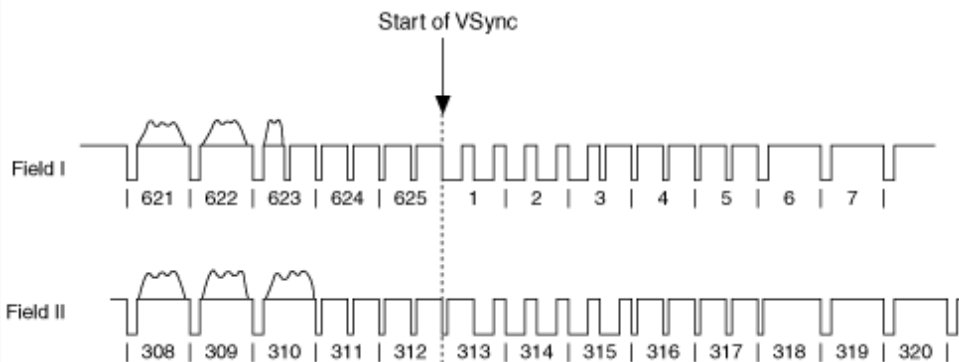
Vertical Blanking and Synchronization Signal for M-NTSC, M-PAL, and 480i



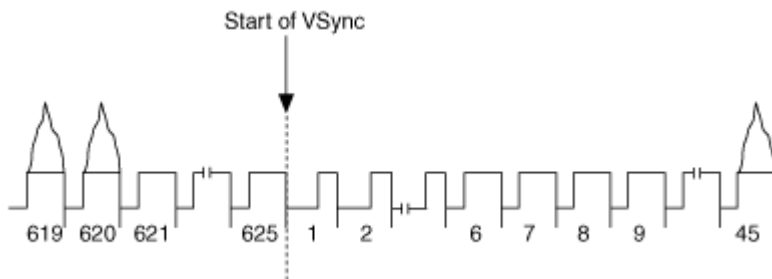
Vertical Blanking and Synchronization Signal for 480p



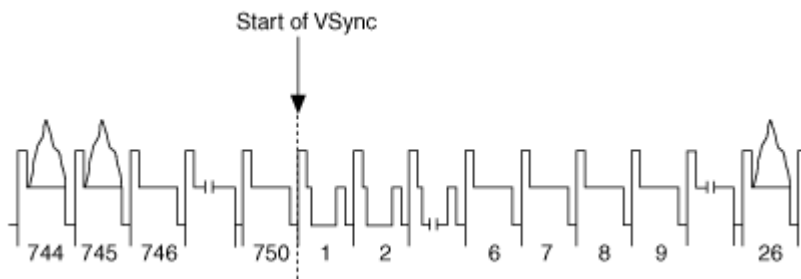
Vertical Blanking and Synchronization Signal for B/G-PAL, SECAM, and 576i



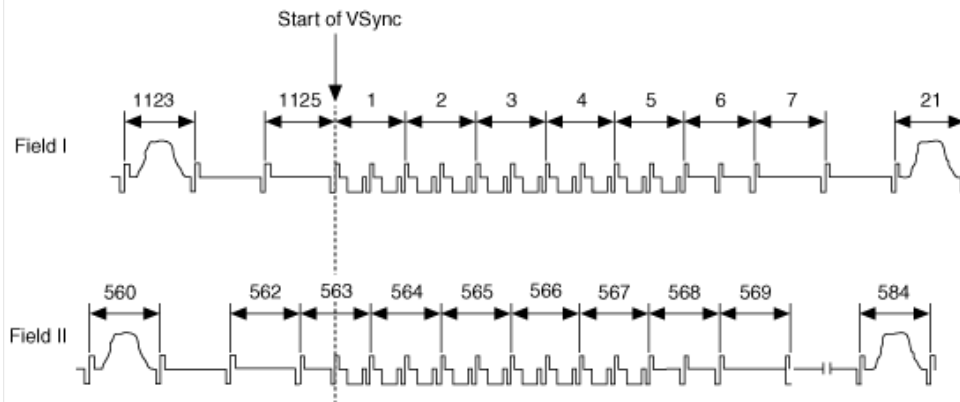
Vertical Blanking and Synchronization Signal for 576p



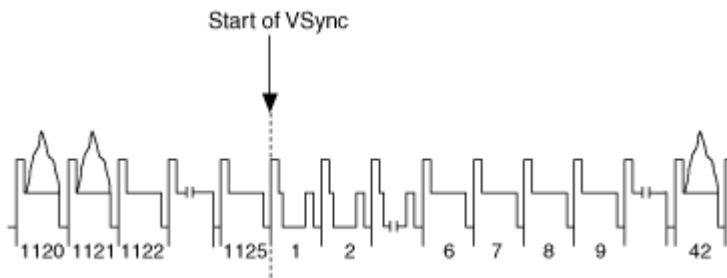
Vertical Blanking and Synchronization Signal for 720p



Vertical Blanking and Synchronization Signal for 1080i



Vertical Blanking and Synchronization Signal for 1080p



Active Image

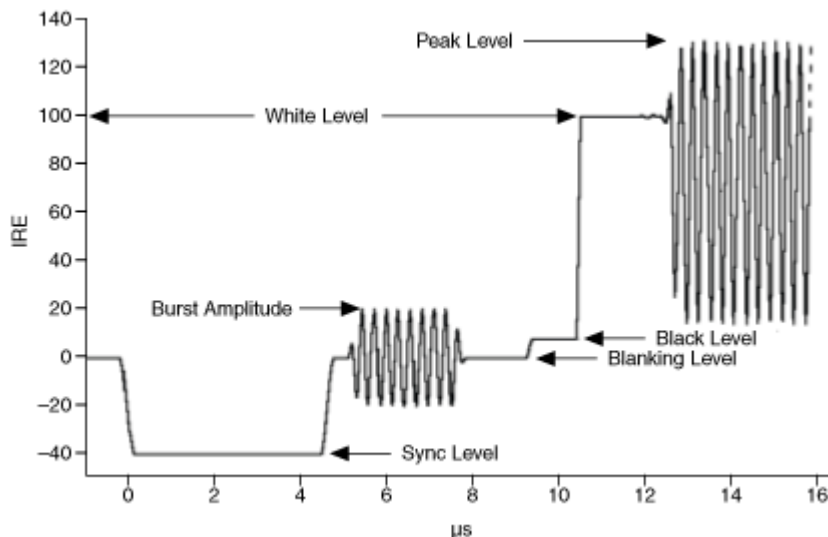
The active video image resulting from the scanning always has an aspect ratio (horizontal/vertical) of 4/3, independent of the video format. The [color composite video signal](#) shows that the scanning process requires some additional room on the left and right sides of each line, as well as on the top and bottom of the active video image region. This additional room includes the synchronization signals, color bursts, and other format-specific information, like the ITS, which are not part of the active video image. Approximately 90% of all the lines and 80% of each line can transmit the active image information. The exact values depend on the video format, as shown in the following table.

Video Format	Lines/Frame	Active Lines	Frame Rate	Line Duration	Active Line Duration
NTSC	525	480/486	29.97 frames/sec	63.55 μ s	52.2 μ s
PAL/SECAM	625	576	25.00 frames/sec	64.00 μ s	52.0 μ s

Active Lines represents the number of lines that are actually used to transmit the image information. For example, only 480 lines out of 525 lines/frame transmit the image information in NTSC. Likewise, on each line, the image information is transmitted only during the active lines sequence, which is shorter than the entire line duration. For example, of 63.55 μ s only 52.2 μ s are the active line duration in NTSC. Frame rate is the scanning speed described in [Scanning Speed](#).

Video Levels

The video levels define the levels and ranges for the different parts of the video signal. The unit used to define video levels is the IRE (Institute of Radio Engineers). The blanking level refers to 0 IRE and the white level refers to +100 IRE. The blanking level, which is the reference level for the video signal (usually 0 V), is different from the black level if a setup is applied to the signal as shown in the following figure.



For NTSC, a setup of 7.5 IRE is usually applied, moving the black level to +7.5 IRE. For PAL and SECAM, the black level is aligned with the blanking level at 0 IRE.

The following table shows the different video levels depending on the video format.

Video Format	Sync Level	Blanking Level	Black Level	White Level	Peak Level	Burst Amplitude
NTSC	-40 IRE	0 IRE	+7.5 IRE	+100 IRE	+120 IRE	20.0 IRE
PAL	-43 IRE	0 IRE	0 IRE	+100 IRE	+133 IRE	21.5 IRE
SECAM	-43 IRE	0 IRE	0 IRE	+100 IRE	+130 IRE	N/A

The analog composite video signal is defined as a voltage source with an output impedance of 75Ω . The sync-to-white level is normally $1 V_{pk-pk}$ when loaded with a 75Ω resistance. Therefore, the unloaded signal is nominally $2 V_{pk-pk}$.

Waveform Measurements

NI-SCOPE waveform measurements offer you another method for fetching data from NI high-speed digitizers. The measurements are processed in software instead of on the digitizer hardware. The results are scalar or array measurements. [Scalar measurements](#) refer to calculations such as rise time and frequency, which take the time-domain data and produce a single value, whereas [array measurements](#) transform the time-domain data into a new waveform, such as an FFT amplitude spectrum.

Related Topics

[Making Waveform Measurements](#)

Reference Levels

Reference levels are high, low, and mid range values that you set to take rise time, fall time, width negative, width positive, duty cycle positive, and duty cycle negative measurements. Reference levels are generally configured in terms of the percentage of the waveform you acquire. By default, the low value is 10% of the waveform, the mid level is 50%, and the high value is 90%.

Use the channel-based attributes to set the reference levels. Refer to [Attributes and Attribute Functions](#) for an overview of attributes.

You can customize how NI-SCOPE configures the percentages used for the reference levels with the Percentage Method attribute. This attribute uses any of the following constants:

- **Min Max**—This method uses the measurement's voltage minimum and voltage maximum as 0% and 100%. These algorithms find the absolute minimum and maximum in the waveform, which is useful for sine waves and triangle waves where the histogram method does not work.
- **Low High**—This method uses the measurement's voltage low and voltage high as 0% and 100%. The voltage low is the voltage of the histogram bin with the most hits below 40% of the waveform's voltage peak-to-peak value. The voltage high is the voltage of the histogram bin with the most hits above 60% of the waveform's voltage peak-to-peak value. These measurements use the last-acquisition histogram method to find the most common high and low voltages, which is useful for ignoring the preshoot and overshoot on a square wave.
- **Base Top**—The voltage base and voltage top measurements correspond to 0% and 100% with this method. These measurements use the last-acquisition histogram method if the most common histogram bin contains a substantial number of the total points. Otherwise, it returns the absolute minimum and maximum values in the waveform. This is a useful default value for most waveform types.

You can also configure the reference levels in terms of voltage with the [Ref Level Units](#) property or the `NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS` attribute.

Hysteresis Percent

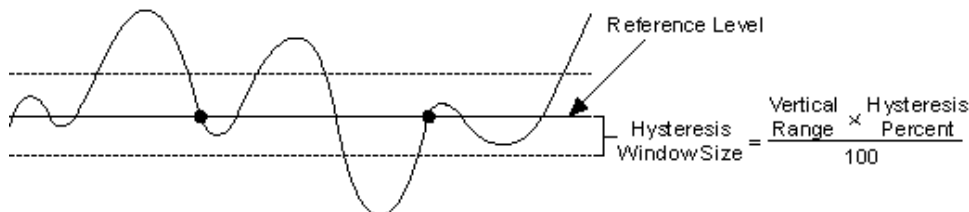
This attribute is used by NI-SCOPE measurements that determine crossing to calculate a measurement. For example, to calculate frequency the algorithm first determines the mid-reference level average for the signal. Next, the algorithm sets up a digital hysteresis around the mid-reference level using the attribute Hysteresis Percent. The algorithm then finds all crossings by counting the number of times the waveform crosses the upper hysteresis level, mid reference level, and lower hysteresis level. Note that the hysteresis levels are calculated based on the vertical range. The algorithm multiplies the hysteresis percent attribute by the vertical range

attribute to determine the upper and lower hysteresis levels. The default value for the hysteresis percent is configured to 2, which means in a 2 V vertical range the algorithm does not see signals/noise that are less than 40 mV.

Measuring Reference-Level Crossings

All scalar measurements involving time use the concept of reference-level crossings, which occur when the waveform voltage crosses the reference level. Linear interpolation accurately estimates the crosspoint times, but noise can create a higher number of level crossings. To minimize the impact of noise, you can create a hysteresis window around crosspoints. To count a crossing, the signal must start outside this window and then pass through the window to the crosspoint. A crossing is not counted again until the signal passes outside the window and then passes through the window to the crosspoint. The following figure shows an exaggerated example of digital hysteresis, where the dots are the real crosspoints after eliminating the spurious crosspoints caused by noise.

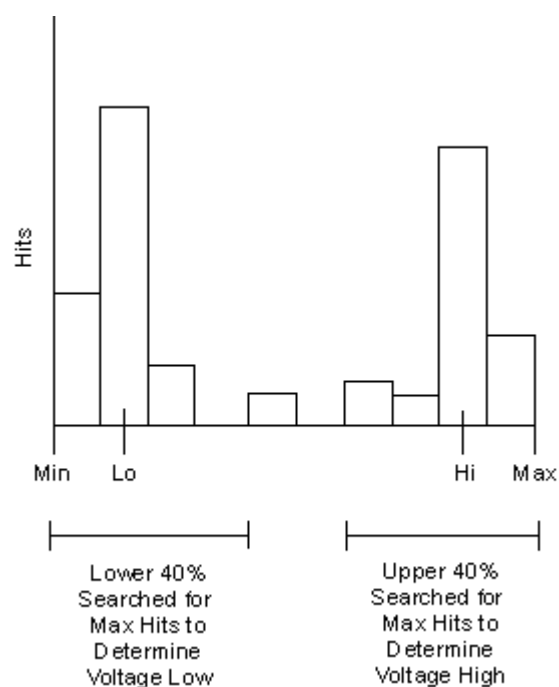
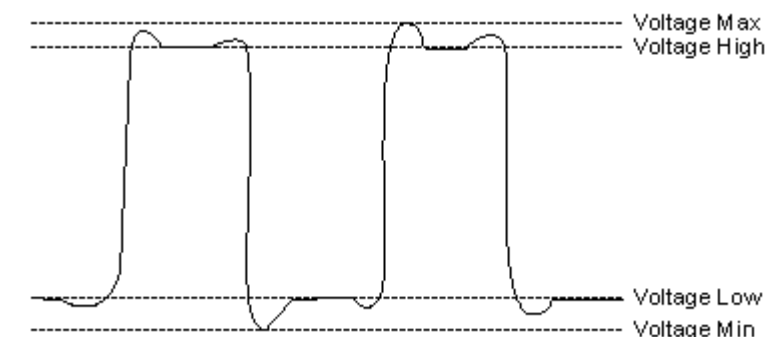
The dots indicate the crosspoints counted for the measurement.



Last-Acquisition Histogram Method

Last-acquisition histogram method is used by the voltage low and voltage high measurements for computing the extrema of a waveform. This method is useful for ignoring overshoot or preshoot in a square waveform. In this method, a voltage histogram is created from the most recent acquisition. The limits of this histogram are set by the minimum and maximum voltages of the acquisition. NI-SCOPE then counts how many samples fall into each bin of the histogram. The resolution is defined by the Last Acquisition Histogram Size attribute, which is 256 bins by default. You can fetch the histogram array using the Fetch Array Measurement function, with Last Acquisition Histogram constant specified as the array measurement function. The following illustrations show an example of a square wave and its resulting histogram. In the histogram the voltage low and voltage high

measurements correspond to the middle value of a bin with the maximum number of hits in the lower 40% or upper 40% respectively.

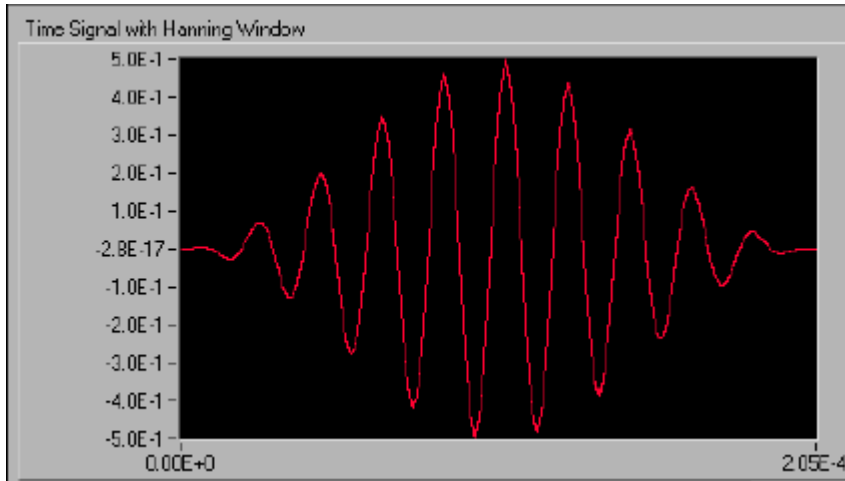


Smoothing Windows

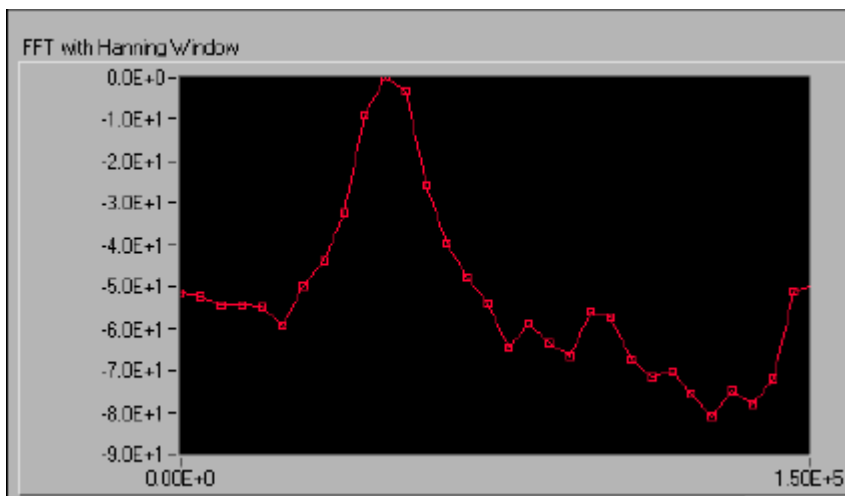
Smoothing windows are a simple means of minimizing [spectral leakage](#) associated with truncated waveforms. Leakage exists because the finite time record of the input signal creates discontinuities when the waveform is extended. The greater the amplitude of these discontinuities, the greater the spectral leakage. A simple way to avoid this is to use smoothing windows when applying an FFT on finite-length data.

You can use smoothing windows to reduce the amplitude of the discontinuities at the boundaries of each period. This technique multiplies the time record by a finite

length window whose amplitude varies smoothly and gradually towards zero at the edges. In the following figure, the original time signal shown in [FFT with Spectral Leakage](#) is windowed using a Hanning window.



The following figure shows that the FFT of this data is significantly less noisy than the non-windowed FFT shown in [FFT with Spectral Leakage](#).



Types of Window Measurements

Choosing the correct window to use requires some prior knowledge of the signal that you are analyzing. The following table shows the different types of windows and the signal types they are appropriate for.

Window Signal	Type and Description	Applications
---------------	----------------------	--------------

Rectangular (no window)	Transient signals that are shorter than the length of the window; truncates a window to within a finite time interval.	Order tracking, system analysis (frequency response measurements) with pseudorandom excitation, separation of two tones with frequencies very close to each other, but with almost equal amplitudes.
Hanning	Transient signals that are longer than the length of the window.	General-purpose applications, system analysis (frequency response measurements) with random excitation.
Hamming	Transient signals that are longer than the length of the window; a modified version of the Hanning window that is discontinuous at the edges.	Often used in speech signal processing.
Blackman	Transient signals; similar to Hanning and Hamming windows but adds one additional cosine term to reduce ripple.	General-purpose applications.
Triangle	Window that is the shape of a triangle.	No special applications.
Flat top	Has the best amplitude accuracy of all the windows but comes at the expense of frequency selectivity.	Accurate single tone amplitude measurements when there are no nearby frequency components.



Note In cases in which you do not have sufficient prior knowledge of the signal, you may need to experiment with different windows to find the best one.

Related Topics

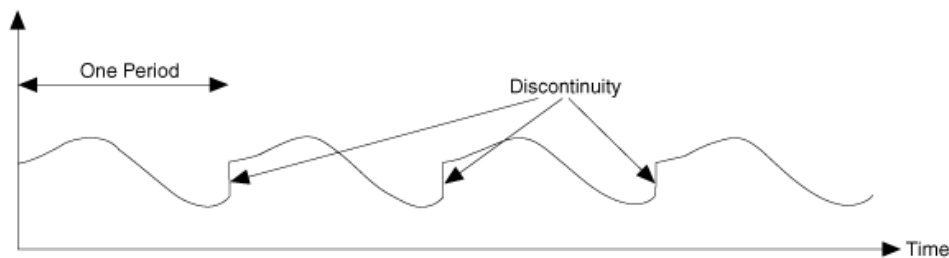
[Spectral Leakage](#)

[FFT without Spectral Leakage](#)

[FFT with Spectral Leakage](#)

Spectral Leakage

When you use the discrete Fourier transform (DFT) or Fast Fourier Transform (FFT) to find the frequency content of a signal, it is assumed that the data that you have is a periodically repeating waveform. In the following figure, the first period shown is the one sampled. Notice that the waveform has not completed an entire cycle, which creates a discontinuity when the waveform is repeated to produce the periodic waveform.



Discontinuities happen when you sample a noninteger number of cycles. These artificial discontinuities turn up as very high frequencies in the spectrum of the signal that were not present in the original signal. These frequencies could be much higher than the Nyquist frequency and are aliased somewhere between 0 and $f_s/2$, where f_s is your sampling rate. The spectrum you get by using a FFT, therefore, is not the actual spectrum of the original signal, but a smeared version. It appears as if the energy at one frequency has leaked out into all the other frequencies. This phenomenon is known as spectral leakage.

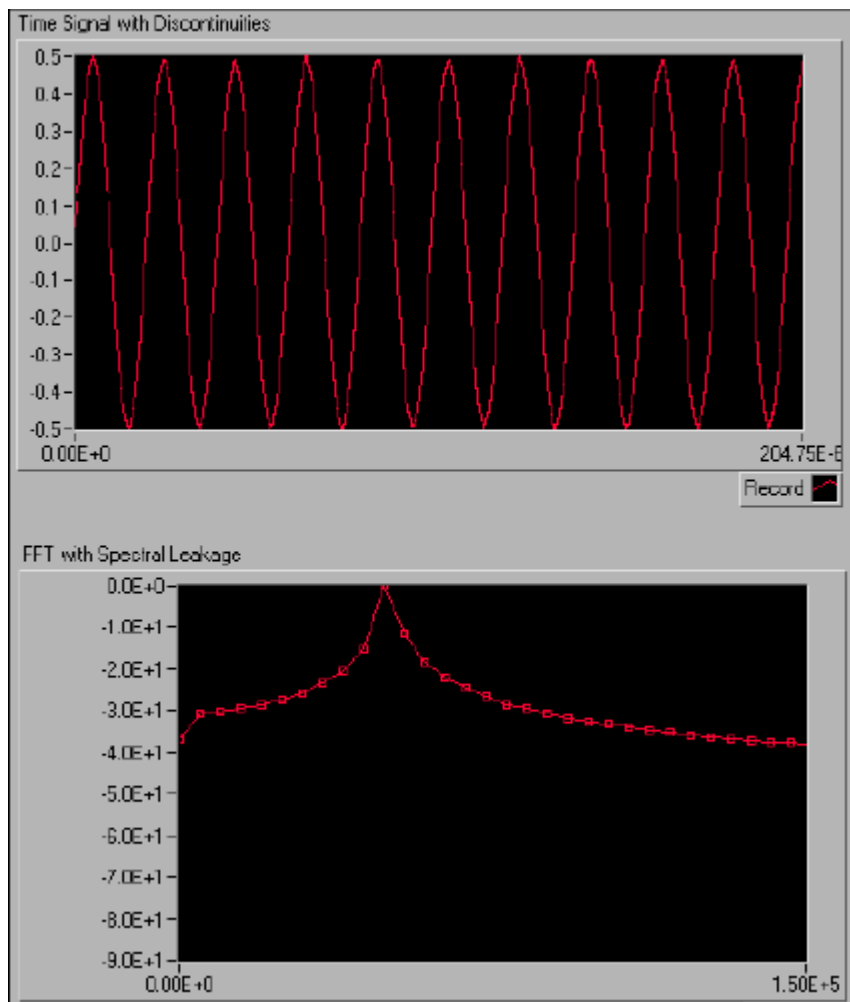
Related Topics

[FFT with Spectral Leakage](#)

[FFT without Spectral Leakage](#)

FFT with Spectral Leakage

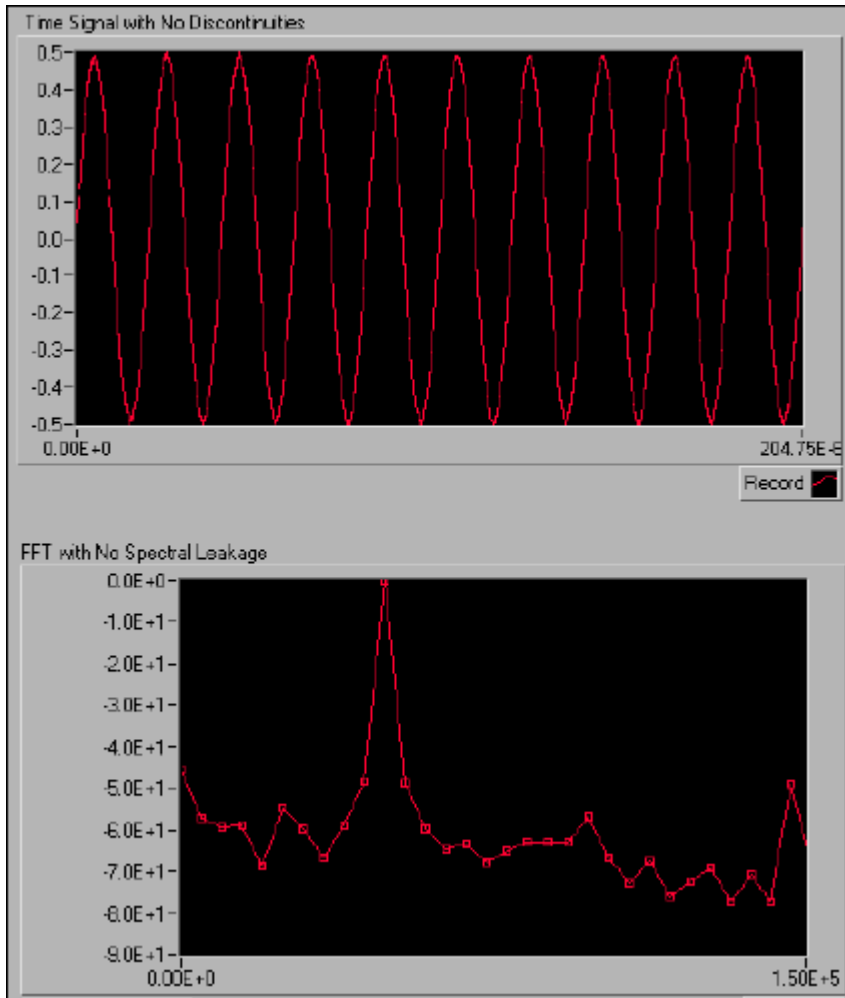
In the following figure, you see the spectral representation when you sample a noninteger number of cycles of the time waveform (namely 10.25). The periodic extension of this signal creates a discontinuity similar to the figure shown in [Spectral Leakage](#).



Notice how the energy is now spread over a wide range of frequencies, so the relative height difference between the FFT peak amplitude and the neighboring bins is reduced. This smearing of the energy is called spectral leakage.

FFT without Spectral Leakage

The following figure shows a sine wave sampled from an NI digitizer and its corresponding FFT amplitude spectrum in decibels. The time-domain waveform has an integer number of cycles (10), so the assumption of periodicity does not create any discontinuities.



Digital Filtering

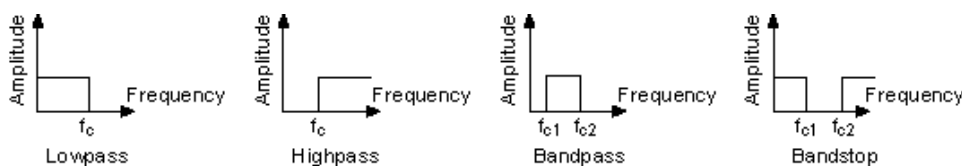
Analog filter design is one of the most important areas of electronic design, but it is often reserved for specialists because it requires advanced mathematical knowledge and understanding of the processes involved in the system affecting the filter. With the digital filters in NI-SCOPE, however, you do not have to be a design expert. NI-SCOPE handles all the design issues, computations, memory management, and actual filtering internally.

Although digital filters have advantages over analog filters, they have disadvantages such as floating-point precision limitations, numerical instability, quantization noise, and frequency warping.

Types of Filters

Filters alter or remove unwanted frequencies. Depending on the frequency range that they either pass or attenuate, they can be classified into the following types:

- A lowpass filter passes low frequencies, but rejects (or attenuates) high frequencies. To specify the cut-off frequency, use the Measurement Filter Cutoff Frequency attribute.
- A highpass filter passes high frequencies, but attenuates low frequencies. To specify the cut-off frequency, use the Measurement Filter Cutoff Frequency attribute.
- A bandpass filter passes a certain band of frequencies and attenuates others. To specify the bandpass filter, use the Measurement Filter Center Frequency attribute and the Filter Width attribute, where the cut-off frequencies are the center frequency plus or minus one-half width.
- A bandstop filter attenuates a certain band of frequencies and passes others. To specify the bandstop filter, use the Measurement Filter Center Frequency attribute and the Measurement Filter Width attribute, where the cut-off frequencies are the center frequency plus or minus one-half width.
- An ideal filter has a gain of one (0 dB) in the passband so that the amplitude of the signal neither increases nor decreases. The stopband (SB) corresponds to that range of frequencies that do not pass through the filter at all and are rejected (attenuated).



Infinite Impulse Response (IIR) Versus Finite Impulse Response (FIR) Filters

Another way to classify filters is by impulse response. An impulse response is the response of a filter to an input that is an impulse ($x[0] = 1$ and $x[i] = 0$ for all $i \neq 0$). The FFT of the filtered impulse response is known as the frequency response of the filter. The frequency response of a filter tells you what the output of the filter is going to be at different frequencies. In other words, it tells you the gain of the filter at different frequencies. For an ideal filter, the gain should be 1 in the passband and 0

in the stopband. So, all frequencies in the passband are passed as is to the output, but there is no output for frequencies in the stopband.

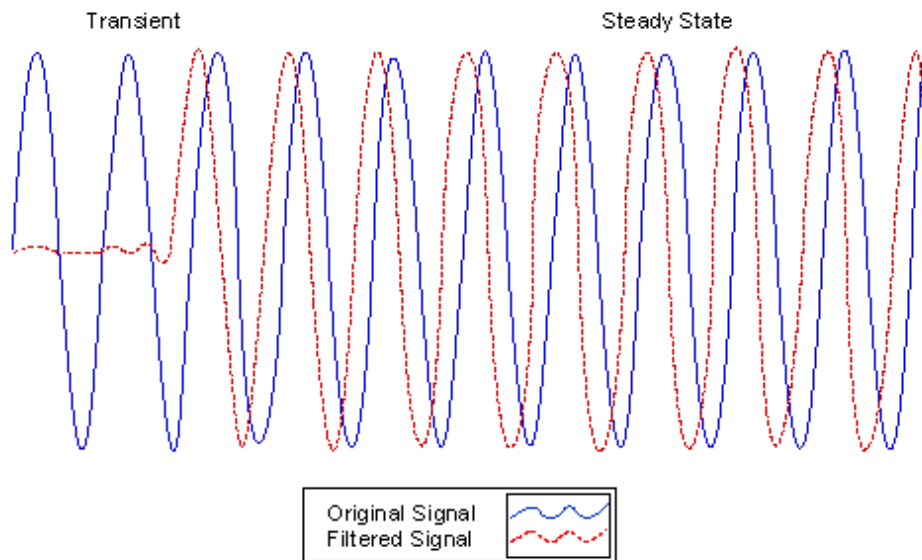
If the impulse response of the filter falls to zero after a finite amount of time, it is known as a FIR filter. However, if the impulse response exists indefinitely, it is known as an IIR filter. Whether the impulse response is finite (that is, whether the filter is FIR or IIR) depends on how the output is calculated. The basic difference between FIR and IIR filters is that for FIR filters, the output depends only on the current and past input values, whereas for IIR filters, the output depends not only on the current and past input values, but also on the past output values. The advantage of digital IIR filters over FIR filters is that IIR filters usually require fewer coefficients to perform similar filtering operations. Thus, IIR filters execute much faster and do not require extra memory, because they execute in place. The disadvantage of IIR filters is that the phase response is nonlinear. If the application does not require phase information, such as amplitude spectrum analysis, IIR filters may be appropriate. You should use FIR filters for those applications requiring linear phase responses.

IIR Filters

Filters are classified as infinite impulse response (IIR) filters if the impulse response exists indefinitely. For IIR filters, the output depends not only on the current and past input values, but also on the past output values.

Truncating Data with IIR Filters

Because IIR filters depend on the output and the input, there is a transient response at the beginning of the filtered data that is invalid, due to the assumptions made at the beginning boundary condition. This is due to the assumption that negative indices in the general IIR difference equation are zero. NI-SCOPE IIR filters remove the user-defined transient portion by deleting the first input size times the Measurement Filter Transient Waveform Percent attribute divided by 100 from the beginning of the filtered data array. This feature is useful when using a combination of processing steps, so the result of an IIR filter excluding the transient portion may be the input of another measurement. The length of the transient response depends on both the filter order and the input waveform, and it may require some trial and error to determine the proper setting for the Measurement Filter Transient Waveform Percent attribute.



Related Topics

[FIR Filters](#)

[Digital Filtering](#)

Types of IIR Filters in NI-SCOPE

NI-SCOPE supports [Butterworth](#), [Chebyshev](#), and [Bessel](#) IIR filters. You can fetch filtered data by calling the `Fetch Array Measurement` function or VI.

Butterworth Filters

NI-SCOPE supports the following attributes for configuring Butterworth filters:

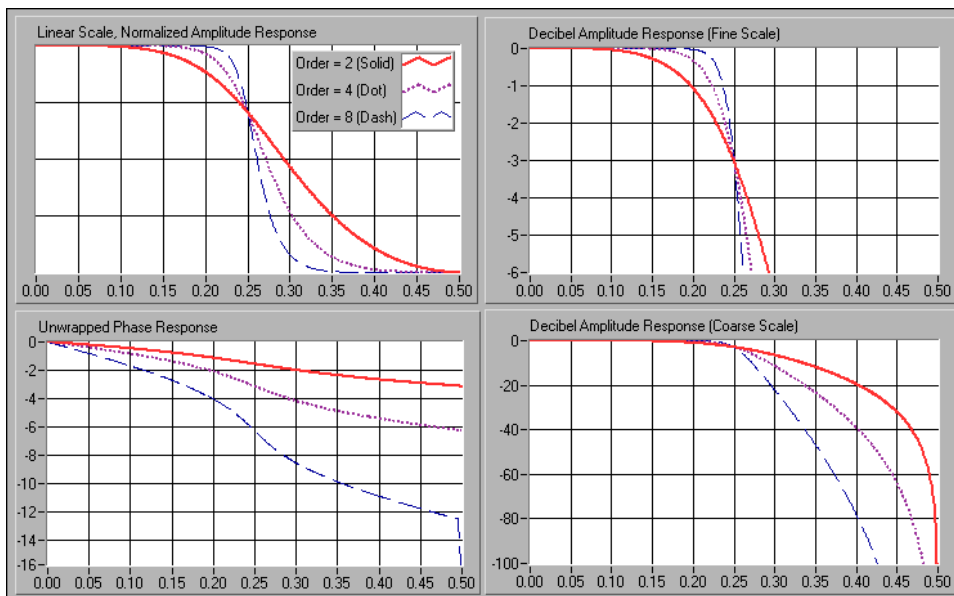
- Measurement Filter Type—lowpass, highpass, bandpass, bandstop
- Measurement Filter Order
- Measurement Transient Waveform Percent

A smooth response at all frequencies and a monotonic decrease from the specified cutoff frequencies characterize Butterworth filters. These filters are maximally flat—the ideal response of unity in the passband and zero in the stopband. The half

power frequency or the -3 dB down frequency corresponds to the specified cutoff frequencies.

Butterworth filters do not always provide a good approximation of the ideal filter response because of the slow rolloff between the passband (the portion of interest in the spectrum) and the stopband (the unwanted portion of the spectrum). The advantage of Butterworth filters is a smooth, monotonically decreasing frequency response. The steepness of the transition is proportional to the filter order, so higher order Butterworth filters approach the ideal lowpass filter response.

The following figure shows the response of a lowpass Butterworth filter. In the figure, the sampling frequency is normalized to 1.0, and the cutoff frequency is 0.25.



Impulse Magnitude and Phase Response Versus Frequency for a Lowpass Butterworth Filter

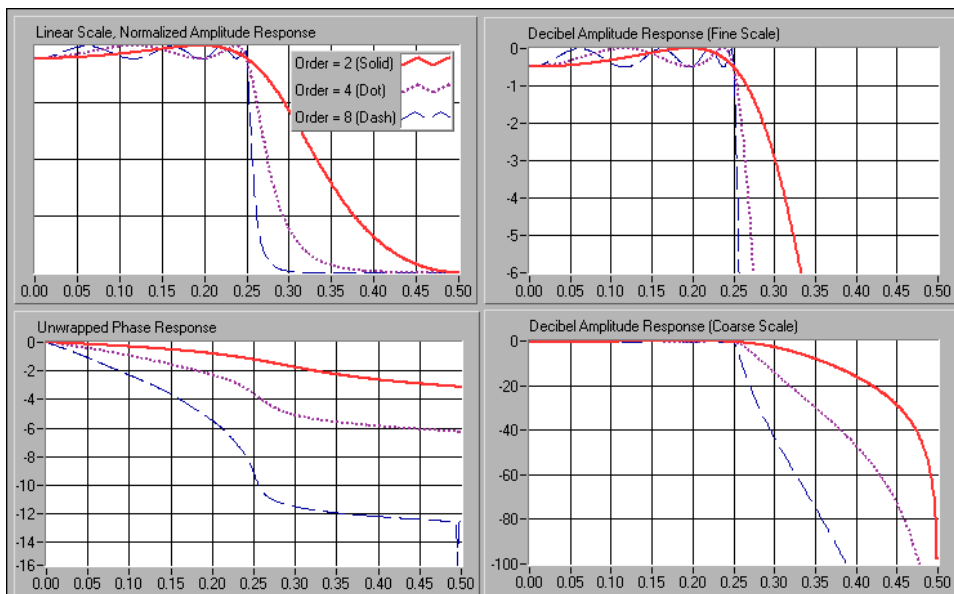
Chebyshev Filters

NI-SCOPE supports the following attributes for configuring Chebyshev filters:

- Measurement Filter Type—lowpass, highpass, bandpass, bandstop
- Measurement Filter Order
- Measurement Passband Filter Ripple in dB
- Measurement Transient Waveform Percent

These attributes minimize peak error in the passband by setting the maximum allowable absolute value of the difference between the ideal filter and the filter response you want (the maximum tolerable error in the passband). Chebyshev filters have an equiripple magnitude response in the passband, monotonically decreasing magnitude response in the stopband, and a sharper rolloff than Butterworth filters. The cutoff frequency for Chebyshev filters is defined as the end of the passband. For example, if you specify a lowpass filter with 1 dB ripple, the passband response from 0 Hz to the cutoff frequency will have 1 dB ripple. At the cutoff frequency, the response will be 1 dB down, and it is monotonically decreasing above the cutoff frequency.

The following figure shows the response of a lowpass Chebyshev filter. Notice that the equiripple response in the passband is constrained by the maximum tolerable ripple error and that the sharp rolloff appears in the stopband. The advantage of Chebyshev filters over Butterworth filters is that Chebyshev filters have a sharper transition between the passband and the stopband with a lower order filter, which produces smaller absolute errors and higher execution speeds.



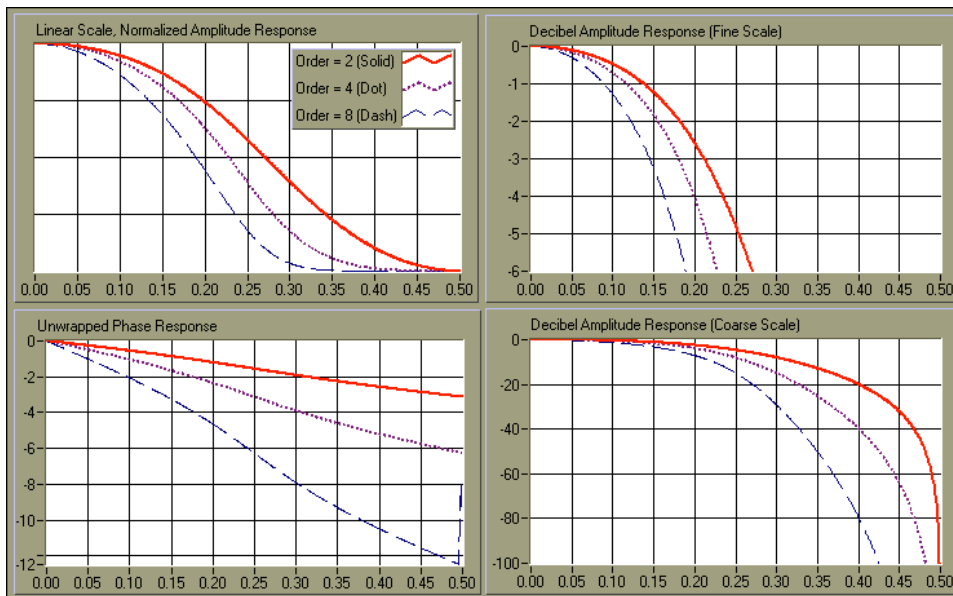
Impulse Magnitude and Phase Response Versus Frequency for a Lowpass Chebyshev Filter

Bessel Filters

NI-SCOPE supports the following attributes for configuring Bessel filters:

- Measurement Filter Type—lowpass, highpass, bandpass, bandstop
- Measurement Filter Order
- Measurement Transient Waveform Percent

You can use Bessel filters to reduce nonlinear phase distortion inherent in all IIR filters. In higher order filters and those with a steeper rolloff, this condition is more pronounced, especially in the transition regions of the filters. Bessel filters have maximally flat response in both magnitude and phase. Furthermore, the phase response in the passband of Bessel filters, which is the region of interest, is nearly linear. Like Butterworth filters, Bessel filters require high-order filters to minimize the error and, for this reason, are not widely used. You can also obtain linear phase response using FIR filter designs. The following figure shows a plot of the response of a lowpass Bessel filter with a sampling frequency normalized to 1.0 and a cutoff frequency of 0.25. Notice that the response is smooth at all frequencies, as well as monotonically decreasing in both magnitude and phase. In addition, notice that the phase in the passband is nearly linear. For Bessel filters, the cutoff frequency specifies the passband, or the region of linear phase response.



Bessel filters have a distinct advantage in the time domain because they exhibit small transient aberrations.

FIR Filters

Finite impulse response (FIR) filters are digital filters that have a finite impulse response. FIR filters are also known as nonrecursive filters, convolution filters, or moving-average (MA) filters because you can express the output of a FIR filter as a finite convolution:

$$y_i = \sum_{k=0}^{n-1} h_k \cdot x_{i-k}$$

where \mathbf{x} represents the input sequence to be filtered, \mathbf{y} represents the output filtered sequence, and \mathbf{h} represents the FIR filter coefficients.

The most important characteristics of FIR filters are the following:

- They can achieve linear phase because of filter coefficient symmetry in the realization.
- They are always stable.
- The filtering function is performed using the convolution and, as such, generally allows associating a delay with the output sequence of $(\mathbf{n}-1)/2$, where \mathbf{n} is the number of filter coefficients or filter taps.

The simplest method for designing linear-phase FIR filters is the window design method. To design a FIR filter by windowing, you start with an ideal frequency response, calculate its impulse response, and then truncate the impulse response to produce a finite number of coefficients. The truncation of the ideal impulse response results in the effect known as the Gibbs phenomenon—oscillatory behavior near abrupt transitions (cutoff frequencies) in the FIR filter frequency response.

You can reduce the effects of the Gibbs phenomenon by smoothing the truncation of the ideal impulse response using a smoothing window. By tapering the FIR coefficients at each end, you can diminish the height of the side lobes in the frequency response. This method widens the main lobe, however, resulting in a wider transition region at the cutoff frequencies. The selection of a window function, then, is similar to the choice between Chebyshev and Butterworth IIR

filters in that it is a trade-off between side lobe levels near the cutoff frequencies and width of the transition region.

Related Topics

[Types of FIR Filters in NI-SCOPE](#)

[IIR Filters](#)

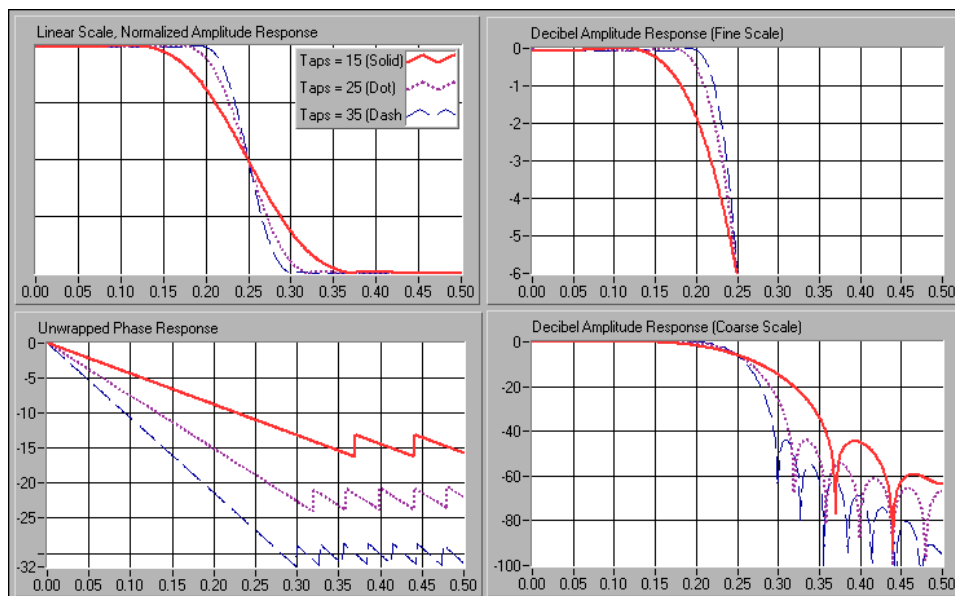
[Digital Filtering](#)

Types of FIR Filters in NI-SCOPE

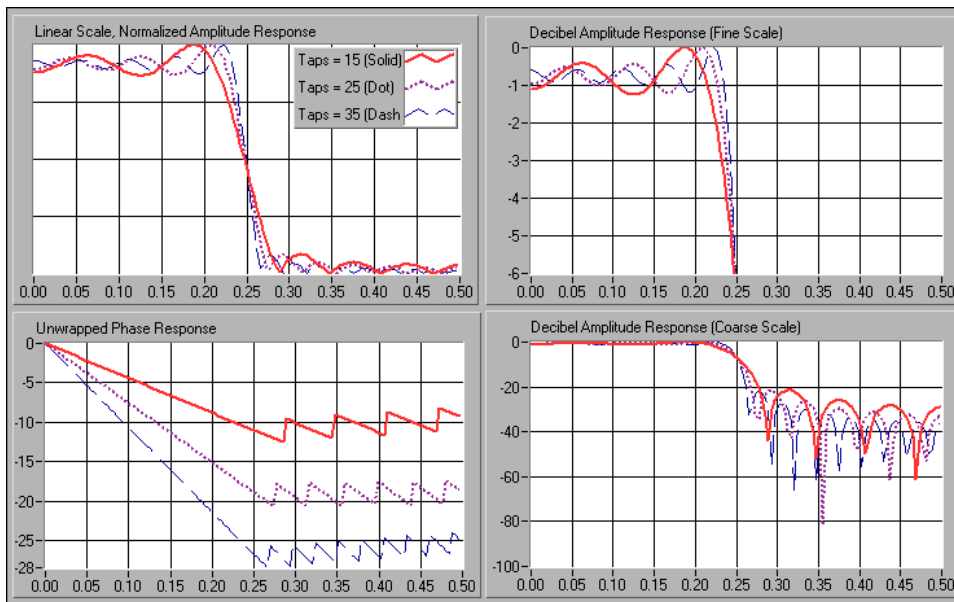
NI-SCOPE supports the following attributes for configuring FIR filters:

- Measurement Filter Type—lowpass, highpass, bandpass, bandstop
- Measurement Filter Taps
- Measurement Filter Window—none, Hanning, Hamming, Triangle, Flat Top, or Blackman

The window type you choose affects the data you acquire. In the following figure, a FIR Filter with a Hanning response window is applied to a signal.



In the following figure, a FIR filter with no window is applied to the same signal, which has been normalized to 1.0 with a cutoff frequency of 0.25.



Related Topics

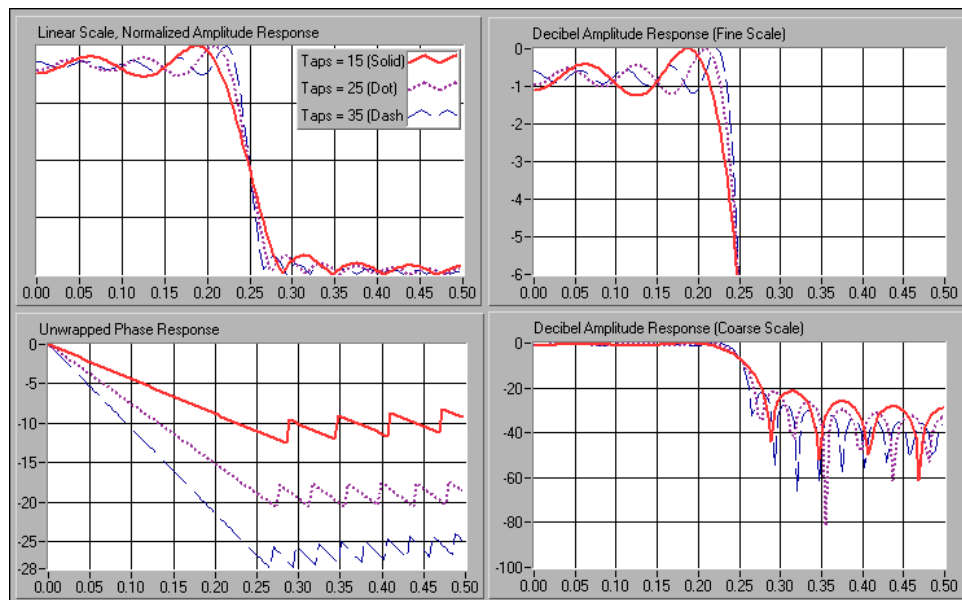
[FIR Filters](#)

[Digital Filtering Overview](#)

FIR No Window Responses

The figure shows the impulse response magnitude and phase response versus frequency for a lowpass windowed FIR filter, where the sampling frequency is normalized to 1.0 and the cutoff frequency is 0.25.

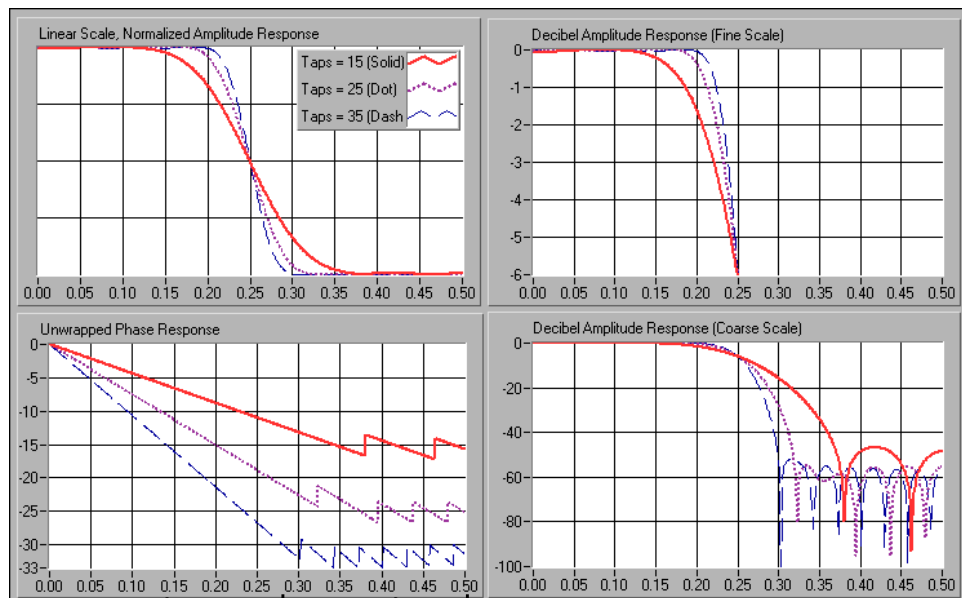
No Window



FIR Hamming Window Responses

The figure shows the impulse response magnitude and phase response versus frequency for a lowpass Hamming windowed FIR filter, where the sampling frequency is normalized to 1.0 and the cutoff frequency is 0.25.

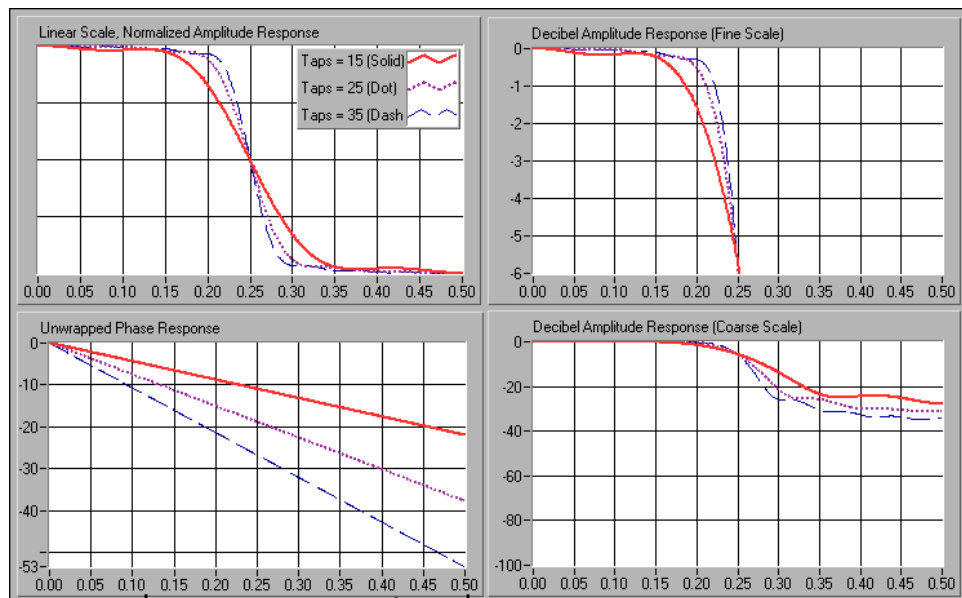
Hamming Window



FIR Triangle Window Responses

The figure shows the impulse response magnitude and phase response versus frequency for a lowpass Triangle windowed FIR filter, where the sampling frequency is normalized to 1.0 and the cutoff frequency is 0.25.

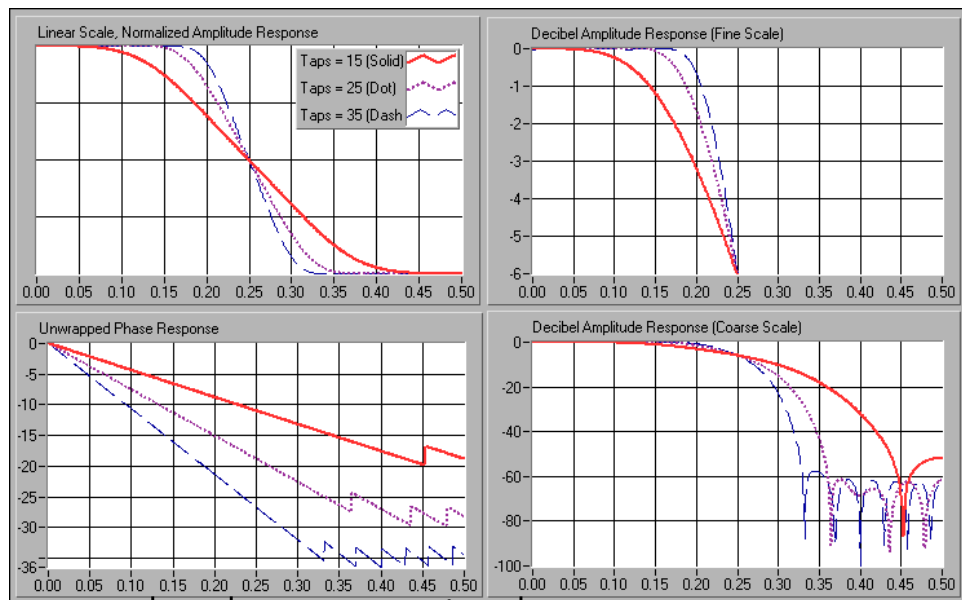
Triangle Window



FIR Flat Top Window Responses

The figure shows the impulse response magnitude and phase response versus frequency for a lowpass Flat Top windowed FIR filter, where the sampling frequency is normalized to 1.0 and the cutoff frequency is 0.25.

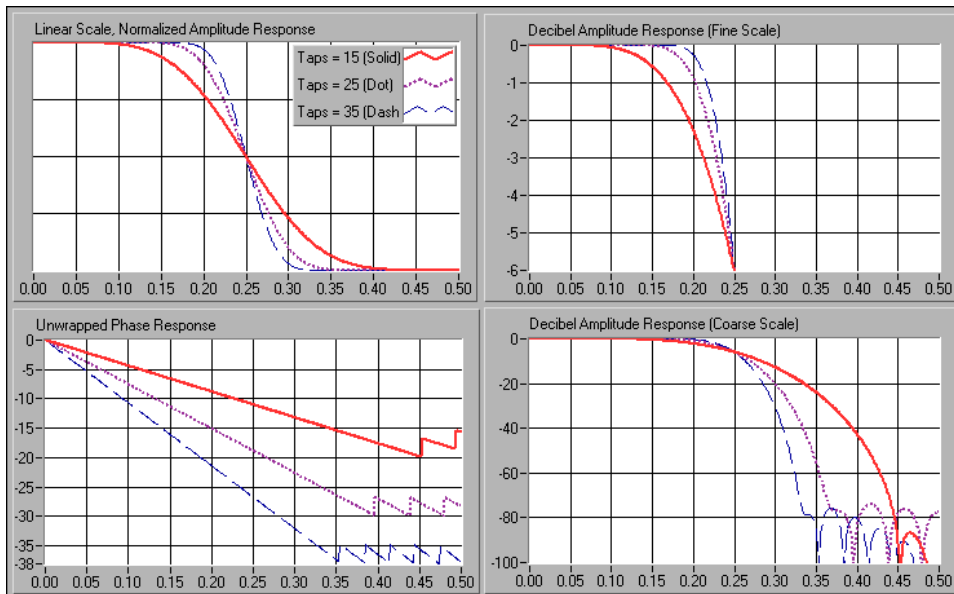
Flat Top Window



FIR Blackman Window Responses

The figure shows the impulse response magnitude and phase response versus frequency for a lowpass Blackman windowed FIR filter, where the sampling frequency is normalized to 1.0 and the cutoff frequency is 0.25.

Blackman Window



Devices

NI high-speed digitizers offer fast setup and data transfer times, fast sampling rates and high bandwidth, deep acquisition memory, multiple-instrument synchronization, and built-in time and frequency domain measurements.

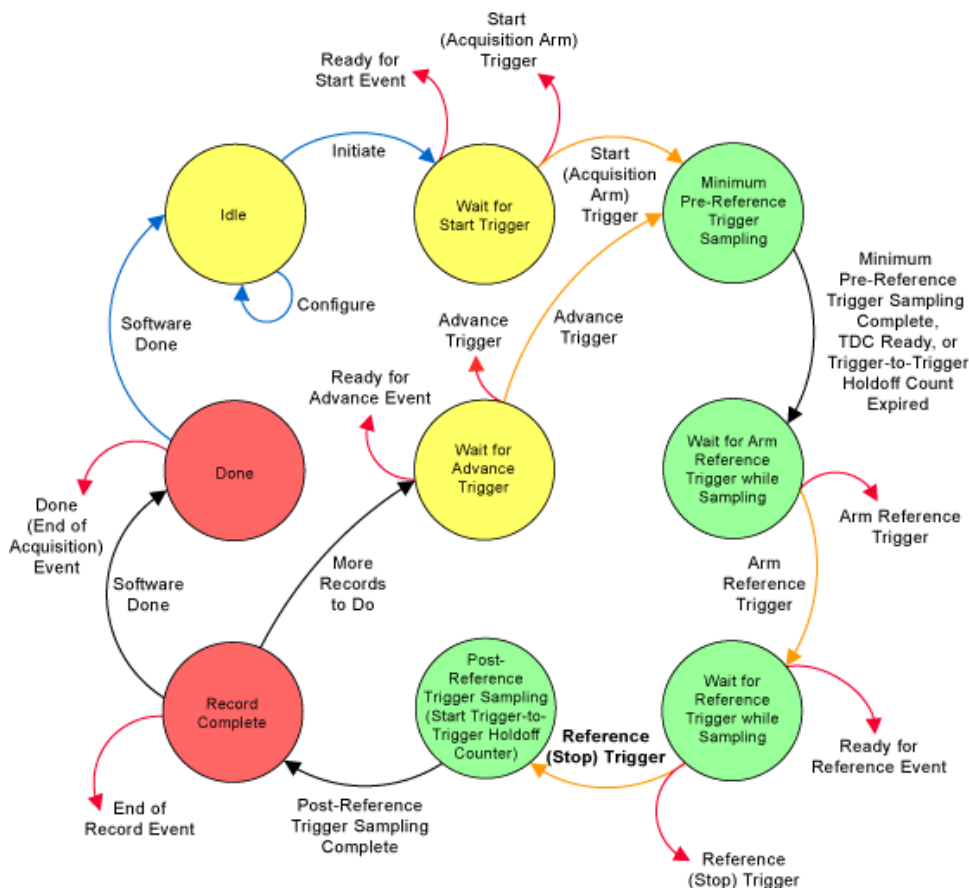
The following sections include an overview of the features and functions of each NI high-speed digitizer or oscilloscope:

- [PXIe/PXI/PCI-5105](#)
- [PXIe-5110](#)
- [PXIe-5111](#)
- [PXIe-5113](#)
- [PXIe/PXI/PCI-5114](#)
- [PXIe/PXI/PCI-5122](#)
- [PXI/PCI-5124](#)
- [USB-5132/5133](#)
- [PXI/PCI-5142](#)

- [PXI/PCI-5152](#)
- [PXI/PCI-5153/5154](#)
- [PXIe-5160/5162](#)
- [PXIe-5163](#)
- [PXIe-5164](#)
- [PXIe-5170/5171](#)
- [PXIe-5172](#)
- [PXIe-5185/5186](#)
- [PXIe-5622](#)
- [PXI-5900](#)
- [PXI/PCI-5922](#)

SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

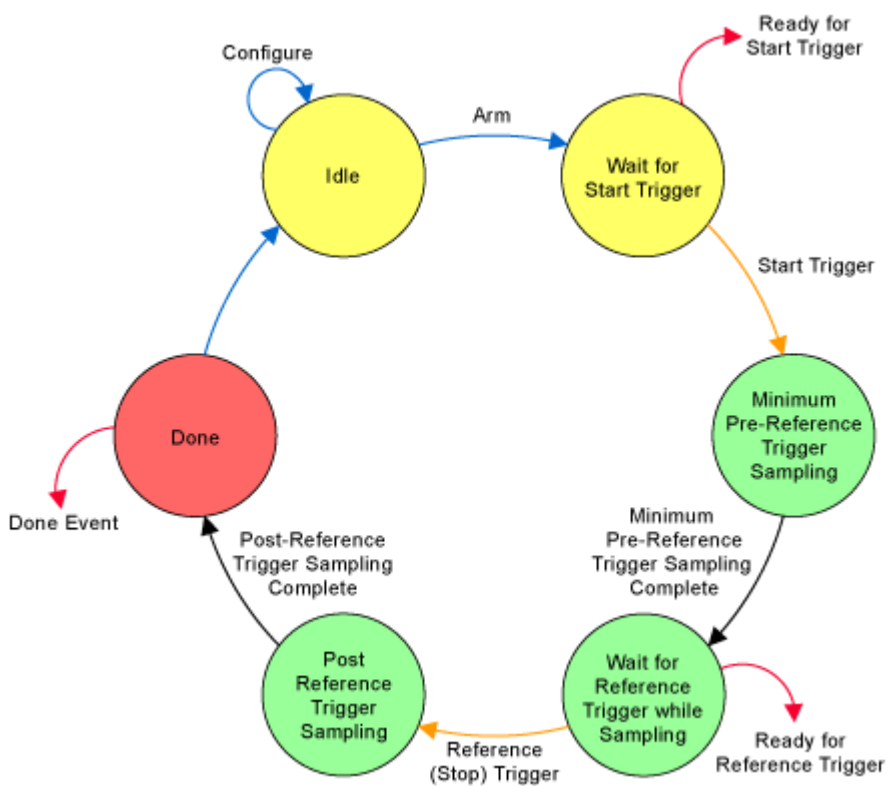
[Triggering](#)

[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

NI USB-5132/5133 Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for the NI USB-5132/5133.



ArrowColor	Indication
Blue	State transitions always caused by software

Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI USB-5132/5133 can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module transitions into this state by receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source. While in this state, the module samples according to the session attributes configured. The module remains in this state until the minimum Pre-Reference Trigger sampling completes. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. When the minimum Pre-Reference Trigger sampling completes, the module transitions out of this state on the next clock cycle.

Wait for Reference—After the module has acquired the minimum pre-reference trigger samples, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger

occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. The module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Done—After the module completes the record, it transitions into this state. Upon entering this state, the module outputs the End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

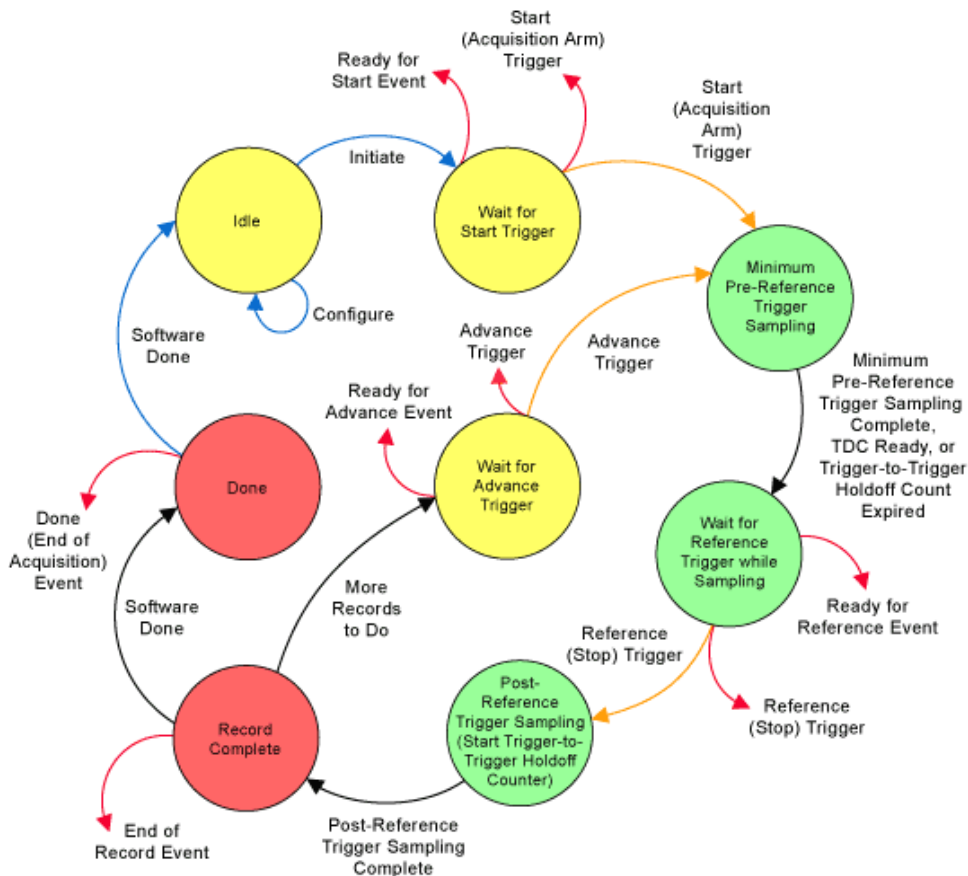
[Triggering](#)

[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

NI 5185/5186 Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for the NI 5185/5186.



ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

The NI 5185/5186 can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When initiate is called on the module, all the

attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Reference Trigger while Sampling—After pre-reference trigger sampling completes, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time,

the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

Features Supported by Device

The following links provide information about which features are supported by each device model.

[SMC-Based Devices and USB Devices](#)

Features Supported by SMC-Based Devices and USB Devices

The tables in this topic show the features supported by the following digitizers:

- NI PXI/PXIe/PCI-5105
- NI PXI/PXIe/PCI-5114
- NI PXI/PXIe/PCI-5122
- NI PXI/PCI-5124
- NI USB-5132
- NI USB-5133
- NI PXI/PCI-5142
- NI PXI/PCI-5152
- NI PXI/PCI-5153
- NI PXI/PCI-5154
- NI PXIe-5160
- NI PXIe-5162
- NI PXIe-5185
- NI PXIe-5186
- NI PXIe-5622
- NI PXI/PCI-5922



Note These devices are configurable in Measurement & Automation Explorer (MAX). Refer to the **NI High-Speed Digitizers Getting Started Guide** for configuration information.

Categories

[Input Configuration](#)

[Timing](#)

[Triggering](#)

[Clocking](#)

[Acquisition](#)

Synchronization Triggers

Miscellaneous

Input Configuration

	NI510 5	NI511 4	NI512 2	NI512 4	NI513 2/5133	NI514 2	NI515 2	NI515 3/5154	NI516 0/5162	NI518 5/5186	NI562 2	NI592 2
Maximum Input Range	±3 V (50 Ω), ±15 V (1 MΩ)	±5 V (50 Ω), ±20 V (1 MΩ)	±5 V (50 Ω), ±10 V (1 MΩ)	±5 V (50 Ω), ±10 V (1 MΩ)	±20 V	±5 V (50 Ω), ±10 V (1 MΩ)	±5 V	±2.5 V	±2.5 V (50 Ω), ±25 V (1 MΩ)	±0.5 V (50 Ω), ±5 V (1 MΩ) ¹	±1.4 V	±5 V
DC Offset	N	Y	Y	Y	Y	Y	Y	N	Y	Y	N	N
Input Impedance	1 MΩ, 50 Ω	1 MΩ, 50 Ω	1 MΩ, 50 Ω	1 MΩ, 50 Ω	1 MΩ	1 MΩ, 50 Ω	1 MΩ, 50 Ω	50 Ω	1 MΩ, 50 Ω	1 MΩ ¹ , 50 Ω	50 Ω	1 MΩ, 50 Ω
Input Coupling	AC (1 MΩ only), DC	AC (1 MΩ only), DC, GND	AC (1 MΩ only), DC, GND	AC (1 MΩ only), DC, GND	AC, DC, GND	AC (1 MΩ only), DC, GND	AC, DC, GND	AC, DC, GND	AC, DC	AC (1 MΩ only) ¹ , DC	AC, GND	AC, DC, GND
3 dB Bandwidth	60 MHz or 24 MHz (analog filter)	125 MHz or 20 MHz (analog filter)	100 MHz, 20 MHz, or 35 MHz (analog filter — default)	150 MHz, 20 MHz, or 60 MHz (analog filter — default)	50 MHz (except 40 mV range), 44 MHz (40 mV range), 20 MHz (analog filter)	100 MHz, 20 MHz, or 35 MHz (analog filter — default)	300 MHz, or 20 MHz (analog filter)	NI 515 3—500 MHz or NI 515 4—1 GHz OR 20 MHz (analog filter)	NI 516 0—500 MHz or 1 MΩ: 300 MHz or Analog filters* NI 516 2—500 MHz or 1 MΩ:	NI 5185—500 MHz or 1 MΩ: 500 MHz ¹ NI 5186—500 MHz or 1 MΩ: 500 MHz ¹	3 MHz to 250 MHz or 162.5 MHz to 212.5 MHz	(0.4 x sample rate); 6 MHz max

										300 M Hz Analog filters*		
										*175 M Hz and 20 MH z analog filters		

¹1 MΩ available on NI 5185 module part number 152962x-0zL and NI 5186 module part number 152961x-0zL, where x is any letter and z is any number. For more information, refer to [NI 5185/5186 Front Panels](#).

Timing

	NI510 5	NI511 4	NI512 2	NI512 4	NI513 2/5133	NI514 2	NI515 2	NI515 3/5154	NI516 0/5162	NI518 5/5186	NI562 2	NI592 2
Maximum Real-Time Sample Rate	60 MS/s	250 MS/s	100 MS/s	200 MS/s	NI 513 2— 50 MS/s NI 513 3— 100 MS/s	100 MS/s	1 GS/s	1 GS/s	1.25 GS/s (all channels)	6.25 GS/s (two channels)	150 MS/s	15 MS/s
Maximum RIS Sample Rate	—	5 GS/s	2 GS/s	4 GS/s	—	2 GS/s	20 GS/s	20 GS/s	NI 5160— 50GS/s NI 5162— 100GS/s	250 GS/s	—	—
Maximum TIS Sample Rate	—	—	—	—	—	—	2 GS/s	2 GS/s	NI 5160— 2.5 GS/s (one or two channels)	12.5 GS/s (one channel)	—	—

									NI 516 2— 2.5 GS/ s (two chann els) 5 GS/s (one chann el)			
Time Stamp	Absolute, Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative	Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative	Absolute, Relative

Triggering

	NI 5105	NI 5114	NI 5122	NI 5124	NI 5132/5133	NI 5142	NI 5152	NI 5153/5154	NI 5160/5162	NI 5185/5186	NI 5622	NI 5922
Reference (Stop) Trigger Types	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software, Video	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software, Video	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software, Video	Immediate, Analog Edge, Digital Edge, Window, Hysteresis, Software	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software, Video	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software	Immediate, Analog Edge, Digital Edge, Hysteresis, Software	Immediate, Analog Edge, Digital Edge, Software	Immediate, Digital Edge, Software	Immediate, Analog Edge, Digital Edge, Hysteresis, Window, Software
PXI Reference (Stop) Trigger Sources	CH 0–7, PFI 1, PXI_Trigger<0..7>, PXI_Star	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	—	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>	—	—	—	CH 0, CH 1, TRIG, PFI<0..1>, PXI_Trigger<0..7>

		PXI Star	PXI Star	PXI Star		PXI Star	PXI Star	PXI Star				PXI Star
PXIe Reference (Stop) Trigger Sources	—	—	CH 0, CH 1, TRIG, PFI<0..7>, PFI<0..7>	—	—	—	—	—	NI 5160/5162 (2 CH) — CH 0, CH 1, TRIG, PFI<0..7>, PFI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>	CH 0, PFI 1, PFI<0..7>	—
PCI Reference (Stop) Trigger Sources	CH 0-7, PFI 1, RTSI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	—	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>	—	—	—	CH 0, CH 1, TRIG, PFI<0..7>, RTSI<0..7>
USB Reference (Stop) Trigger Sources	—	—	—	—	CH 0, CH 1, PFI 1	—	—	—	—	—	—	—
Trigger Coupling (E	—	AC, DC, LF Rej	AC, DC, LF Rej	AC, DC, LF Rej	—	AC, DC, LF Rej	AC, DC, LF Rej	DC, LF Rej	NI 5160/5162 (2 CH)	DC	—	DC

External TRIG Input)		ect, HF Reject, AC+HF Reject	ect, HF Reject, AC+HF Reject	ect, HF Reject, AC+HF Reject		ect, HF Reject, AC+HF Reject	ect, HF Reject, AC+HF Reject	HF Reject	— AC, DC, LF Reject, HF Reject, AC+HF Reject			
Trigger Coupling (Analog Input Channel)	Same as input channel	Same as input channel, HF Reject, LF Reject	Same as input channel, HF Reject, LF Reject	Same as input channel, HF Reject, LF Reject	Same as input channel	Same as input channel, HF Reject, LF Reject	Same as input channel, HF Reject, LF Reject	Same as input channel, HF Reject, LF Reject	Same as input channel, HF Reject, LF Reject	Same as input channel	—	Same as input channel
Trigger Hold off	Y	Y	Y	Y	—	Y	Y	Y	Y	Y	Y	Y
Trigger Delay	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
External TRIG Input Impedance	—	1 MΩ	1 MΩ	1 MΩ	—	1 MΩ	1 MΩ	2.25 kΩ	NI 5160/5162 (2 CH) — 50 Ω or 1 MΩ	50 Ω	—	1 MΩ

Clocking

	NI5105	NI5114	NI5122	NI5124	NI5132/5133	NI5142	NI5152	NI5153/5154	NI5160/5162	NI5185/5186	NI5622	NI5922
PXI/PXIe Reference Clock Input	PFI 1, PXI_CLK10	CLK IN, PXI_CLK10	CLK IN, PXI_CLK10	CLK IN, PXI_CLK10	—	CLK IN, PXI_CLK10	PFI 0, PXI_CLK10	PFI 0, PXI_CLK10	CLK IN, PXI_CLK10	REF CLK, PXIe_CLK100	CLK IN, PXIe_CLK100	CLK IN, PXI_CLK10

t Sources												
PCI Reference Clock Input Sources	PFI 1, RTSI Clock (RTSI 7)	CLK IN, RTSI Clock (RTSI 7)	CLK IN, RTSI Clock (RTSI 7)	CLK IN, RTSI Clock (RTSI 7)	—	CLK IN, RTSI Clock (RTSI 7)	PFI 0, RTSI Clock (RTSI 7)	PFI 0, RTSI Clock (RTSI 7)	—	—	—	CLK IN, RTSI Clock (RTSI 7)
Reference Clock Frequency	1 MHz to 20 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)	—	1 MHz to 20 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)	10 MHz	10 MHz and 100 MHz	1 MHz to 100 MHz (1 MHz increments)	1 MHz to 20 MHz (1 MHz increments)
PXI/PXle Reference Clock Outputs	PFI 1	PFI<0..1>, PXI_Trig<0..6>	CLK OUT, PFI<0..1>, PXI_Trig<0..6>	CLK OUT, PFI<0..1>, PXI_Trig<0..6>	—	CLK OUT, PFI<0..1>, PXI_Trig<0..6>	PFI 1, PXI_Trig<0..6>	PFI 1, PXI_Trig<0..6>	CLK OUT	—	CLK OUT	CLK OUT, PFI<0..1>, PXI_Trig<0..6>
PCI Reference Clock Outputs	PFI 1, RTSI Clock (RTSI 7)	PFI<0..1>, RTSI Clock (RTSI 7)	CLK OUT, PFI<0..1>, RTSI Clock (RTSI 7)	CLK OUT, PFI<0..1>, RTSI Clock (RTSI 7)	—	CLK OUT, PFI<0..1>, RTSI Clock (RTSI 7)	RTSI Clock (RTSI 7)	RTSI Clock (RTSI 7)	—	—	—	CLK OUT, PFI<0..1>, RTSI Clock (RTSI 7)
External Sample Clock Sources	PFI 1, PXI Star	CLK IN	CLK IN, PXI Star	CLK IN, PXI Star	PFI 1	CLK IN, PXI Star	PFI 0	PFI 0	CLK IN	CLK IN	CLK IN	—
External Sample Clock	4 MHz to 65 MHz	50 MHz to 250 MHz	30 MHz to 105 MHz	50 MHz to 210 MHz	1 MHz to 50 MHz	30 MHz to 105 MHz	350 MHz to 1 GHz	350 MHz to 1 GHz	1.25 GHz to 2.5 GHz	1.6 GHz to 3.125 GHz	20 MHz to 150 MHz	—

Frequency					(NI 513 2), 1 MHz to 100 M Hz (NI 513 3)							
Sample Clock Outputs	PFI 1 ²	—	CLK OUT, PFI<0..1>, RTSI<0..6>	CLK OUT, PFI<0..1>, RTSI<0..6>	—	CLK OUT, PFI<0..1>	—	—	—	—	CLK OUT	—

²Exporting sample clock valid at full rate only.

Acquisition

	NI5105	NI5114	NI5122	NI5124	NI5132/5133	NI5142	NI5152	NI5153/5154	NI5160/5162	NI5185/5186	NI5622	NI5922
Acquisition Modes	Normal	Normal	Normal	Normal	Normal	Normal, DDC	Normal	Normal	Normal	Normal	Normal, DDC	Normal, Flex Res
Multiple Records	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
Continuous Acquisition	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Synchronization Triggers

	NI5105	NI5114	NI5122	NI5124	NI5132/5133	NI5142	NI5152	NI5153/5154	NI5160/5162	NI5185/51863	NI5622	NI5922
PXI Arm Reference	PFI 1, PFI<0..7> PFI<0..7>	PFI<0..1>, PFI<0..1> PFI<0..1>	PFI<0..1>, PFI<0..1> PFI<0..1>	PFI<0..1>, PFI<0..1> PFI<0..1>	—	PFI<0..1>, PFI<0..1> PFI<0..1>	PFI<0..1>, PFI<0..1> PFI<0..1>	PFI<0..1>, PFI<0..1> PFI<0..1>	—	—	—	PFI<0..1>, PFI<0..1> PFI<0..1>

e Trigger Sources	>, PXI Star	ig<0..7>, PXI Star	ig<0..7>, PXI Star	ig<0..7>, PXI Star		ig<0..7>, PXI Star	ig<0..7>, PXI Star	ig<0..7>, PXI Star				g<0..7>, PXI Star
PXIe Arm Reference Trigger Sources	—	—	PFI<0..1>, PFI_Trig<0..7>	—	—	—	—	—	PFI<0..1>, PFI_Trig<0..7>	—	PFI 1, PFI_Trig<0..7>	—
PCI Arm Reference Trigger Sources	PFI 1, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	—	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	—	—	—	PFI<0..1>, RTSI<0..7>
Record Advance Trigger Sources	PFI 1, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	—	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, PFI_Trig<0..7>	PFI_Trig<0..6>	PFI 1, PFI_Trig<0..7>	PFI<0..1>, RTSI<0..7>, PFI Star
PXI Start Trigger (Acquisition Arm) Sources	PFI 1, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	—	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	PFI<0..1>, RTSI<0..7>, PFI Star	—	—	—	PFI<0..1>, RTSI<0..7>, PFI Star
PXIe Start Trigger (Acquisition Arm) Sources	—	—	PFI<0..1>, RTSI<0..7>	—	—	—	—	—	PFI<0..1>, PFI_Trig<0..7>	PFI_Trig<0..6>	PFI 1, PFI_Trig<0..7>	—
PCI Start Trigger (Acquisition)	PFI 1, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	—	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	PFI<0..1>, RTSI<0..7>	—	—	—	PFI<0..1>, RTSI<0..7>

tion Arm) Sources												
USB Start Trigger (Acquisition Arm) Sources	—	—	—	—	PFI 1	—	—	—	—	—	—	—
Reference (Stop) Trigger Outputs	PFI 1, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI 1	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, PXI_Trig<0..6>	PXI_Trig<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>
End of Record Outputs	PFI 1, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	—	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, PXI_Trig<0..6>	PXI_Trig<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>
End of Acquisition Outputs	PFI 1, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI 1	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, PXI_Trig<0..6>	PXI_Trig<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>
Start Trigger Outputs	PFI 1, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI 1	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, PXI_Trig<0..6>	PXI_Trig<0..6>	PFI<0..1>, RTSI<0..6>	PFI<0..1>, RTSI<0..6>

³The NI 5185/5186 can simultaneously export/import a maximum of 4 triggers/events.

Miscellaneous

	NI5105	NI5114	NI5122	NI5124	NI5132/5133	NI5142	NI5152	NI5153/5154	NI5160/5162	NI5185/5186	NI5622	NI5922
Self-Calibration	Y	Y	Y	Y	Y(DC Offset only)	Y	Y	Y	Y	Y	Y	Y

Probe Compensation	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	PFI 1 (1 kHz)	—	—	—
LabVIEW Real-Time Support (PXI/PXIe Only)	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
Onboard Signal Processing Support	N	N	N	N	N	Y	N	N	N	N	Y	N
PXI Memory per Channel	16, 128, or 512 MB ⁴	8, 64, or 256 MB	8, 32, 256, or 512 MB	8, 32, 256, or 512 MB	—	64 or 256 MB	8, 64, or 256 MB	8, 64, or 256 MB	—	—	—	8, 32, or 256 MB
PXIe Memory per Channel	—	—	8, 64, or 256 MB	—	—	—	—	—	up to 2GB ⁴	32 MB or 1 GB ⁴	64 or 256 MB	—
PCI Memory per Channel	16, 128, or 512 MB ⁴	8, 64, or 256 MB	8, 32, 256, or 512 MB	8, 32, 256, or 512 MB	—	64 or 256 MB	8, 64, or 256 MB	8, 64, or 256 MB	—	—	—	8, 32, or 256 MB
USB Memory per Channel	—	—	—	—	8 MB or 64 MB ⁴	—	—	—	—	—	—	—
Sample Width	16 bits	8 bits	8 bits, 16 bits (default)	8 bits, 16 bits (default)	8 bits	8 bits, 16 bits (default)	8 bits	8 bits	8 bits, 16 bits (default)	8 bits	8 bits, 16 bits (default)	32 bits (default); configurable

Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment

Manual adjustment significantly reduces skew between instruments and enhances synchronization repeatability, which results in greater coherence of multi-instrument measurements.

For background on manual adjustment, refer to [Manual Adjustment of NI-TClk Synchronization](#).

Before beginning this procedure, complete the following:

- Self-calibrate all instruments in your test system.
- Obtain a reference signal with as low skew and jitter as possible.



Tip The lower the skew and jitter of your reference signal, the better the synchronization you can achieve between instruments using manual adjustment.

This procedure applies to:

- Manual adjustment of NI oscilloscopes that support NI-TClk synchronization
- Synchronizing multiple NI oscilloscopes of a single bus type and model number (for example, all PXI-5922 modules, but not PXI-5922 and PCI-5922 modules)

Complete the following steps to perform manual adjustment for your oscilloscopes.

Set Up

1. Connect the reference signal to all instruments you are synchronizing.




Tip For best results, use cables of the same model and of identical length for each connection.

2. Configure an acquisition as appropriate for your reference signal.
3. Synchronize the instruments in your session using the NI-TClk API.
 1. (LabVIEW, LabVIEW NXG only) Call NI-SCOPE Get Session Reference to generate a reference from NI-SCOPE that NI-TClk can use.
 2. Call NI-TClk Configure for Homogeneous Triggers.
 3. Call NI-TClk Synchronize.

4. Call NI-TClk Initiate.

Calculate Adjustment

4. Fetch data from the instruments and measure the skew and jitter between the waveforms each instrument is acquiring, for example, by performing a point-by-point average.
5. Use any of the adjustment techniques applicable to your instrument to reduce the skew and jitter you measured:


Instrument	Adjustment Technique	Description		When to Use
<ul style="list-style-type: none"> ▪ _PXIe/PXI/PCI-5105 ▪ _PXIe/PXI/PCI-5114 ▪ _PXIe/PXI/PCI-5122 ▪ _PXI/PCI-5124 ▪ _PXI/PCI-5142 ▪ _PXI/PCI-5152 ▪ _PXI/PCI-5153 ▪ _PXI/PCI-5154 ▪ _PXIe-5622 ▪ _PXI/PCI-5922 	NI-SCOPE Adjust Sample Clock Relative Delay function	<p>Offsets the sample clock from the reference clock by the specified amount.</p> <p>Reduces skew between instruments.</p>	<ul style="list-style-type: none"> ▪ Adjustment range: ± 1 Sample Clock Period ▪ Adjustment unit: seconds <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  Tip You can call this function multiple times to implement cumulative delay. </div>	Primary technique for reducing skew and jitter for supported instruments. ¹
<ul style="list-style-type: none"> ▪ _PXIe-5110 ▪ _PXIe-5111 ▪ _PXIe-5113 ▪ _PXIe-5163 ▪ _PXIe-5164 ▪ _PXIe-5170 ▪ _PXIe-5171 ▪ _PXIe-5172 	NI-SCOPE Absolute Sample Clock Offset property	<p>Offsets the sample clock from the reference clock by the specified amount.</p> <p>Reduces skew between instruments and improves synchronization repeatability.</p>	<ul style="list-style-type: none"> ▪ Adjustment range: ± 0.5 Sample Clock Period ▪ Adjustment unit: seconds 	Primary technique for reducing skew and jitter for supported instruments.

Instrument	Adjustment Technique	Description	When to Use
All oscilloscopes that support NI-TClk synchronization except the following:	NI-TClk Sample Clock Delay property	Offsets the sample clock of an NI-TClk session relative to the other synchronized NI-TClk sessions. Reduces skew between instruments.	To apply offset when adjustment beyond the range of other techniques is needed. Sole technique for reducing skew for the PXIe-5160 and PXIe-5162.
<ul style="list-style-type: none"> ▪ PXIe-5185² ▪ PXIe-5186² 		<ul style="list-style-type: none"> ▪ Adjustment range: $\pm\infty$ ▪ Adjustment unit: seconds 	

- Repeat steps 4 and 5 until the measured skew and/or jitter are as low as possible.

Apply Adjustment

- Depending on your adjustment technique(s) from step 5, save the adjustment you used to minimize skew and/or jitter.

Adjustment Technique	Description
NI-SCOPE Adjust Sample Clock Relative Delay	<p>The NI-SCOPE Oscillator Phase DAC Value property converts the time offset you applied with NI-SCOPE Adjust Sample Clock Relative Delay into a value that can apply the manual adjustment across sessions and improve synchronization repeatability.</p> <ol style="list-style-type: none"> Add the NI-SCOPE Oscillator Phase DAC Value property to your program and read its value. Save the Oscillator Phase DAC Value you read. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note You do not need to include the NI-SCOPE Adjust Sample Clock Relative Delay function in your program after you have applied the adjustment with the NI-SCOPE Oscillator Phase DAC Value property.</p> </div>
All other adjustment techniques	Save the value of each property.

- Write the saved value(s) of the property or properties you used prior to calling the NI-TClk Synchronize function in other sessions when using the test system

in an identical physical configuration. Writing the manual adjustment property or properties to a session applies the improved synchronization to the session, regardless of the signals the test system is acquiring.

Reperform manual adjustment for all supported instruments in the following situations:

- If any chassis, instruments, cables, or other physical components of your test system are moved, replaced, or removed
- If the onboard temperature of any module in the test system changes by $\pm 1^\circ$ C since the last self-calibration
- If the trigger type, filtering characteristics, or any clocking parameters of your application are changed

Reperform manual adjustment for specific instruments in these additional situations:

- PXIe-5185 and PXIe-5186 only:
 - If any instrument is reset
 - If any instrument is self-calibrated
 - If the host computer or a chassis within the test system is rebooted
- PXI-5922 and PCI-5922 only: If any channel-based properties are changed

Related information

[NI-SCOPE Get NI-TClk Session Reference](#)

[NI-TClk Configure for Homogenous Triggers](#)

[NI-TClk Synchronize](#)

[NI-TClk Initiate](#)

[NI-SCOPE Adjust Sample Clock Relative Delay](#)

[NI-SCOPE Absolute Sample Clock Offset property](#)

[NI-TClk Sample Clock Delay property](#)

[NI-SCOPE Oscillator Phase DAC Value property](#)

¹ Must be used in conjunction with the NI-SCOPE Oscillator Phase DAC Value property.

² PXIe-5185 and PXIe-5186 only: These oscilloscopes have limited support for manual adjustment. To calculate an adjustment for these oscilloscopes, you must

set the NI-SCOPE Oscillator Phase DAC Value property and determine the effect it has on measured skew and jitter experimentally.

- `_Reduces skew between instruments and improves synchronization repeatability`
- `_Valid values: [0, 65535]`

This is the sole technique for reducing skew and jitter for the PXIe-5185 and PXIe-5186.

NI 5105

The NI 5105 features eight channels, which is ideal for use in large channel-count systems.

Features

The NI 5105 has the following features:

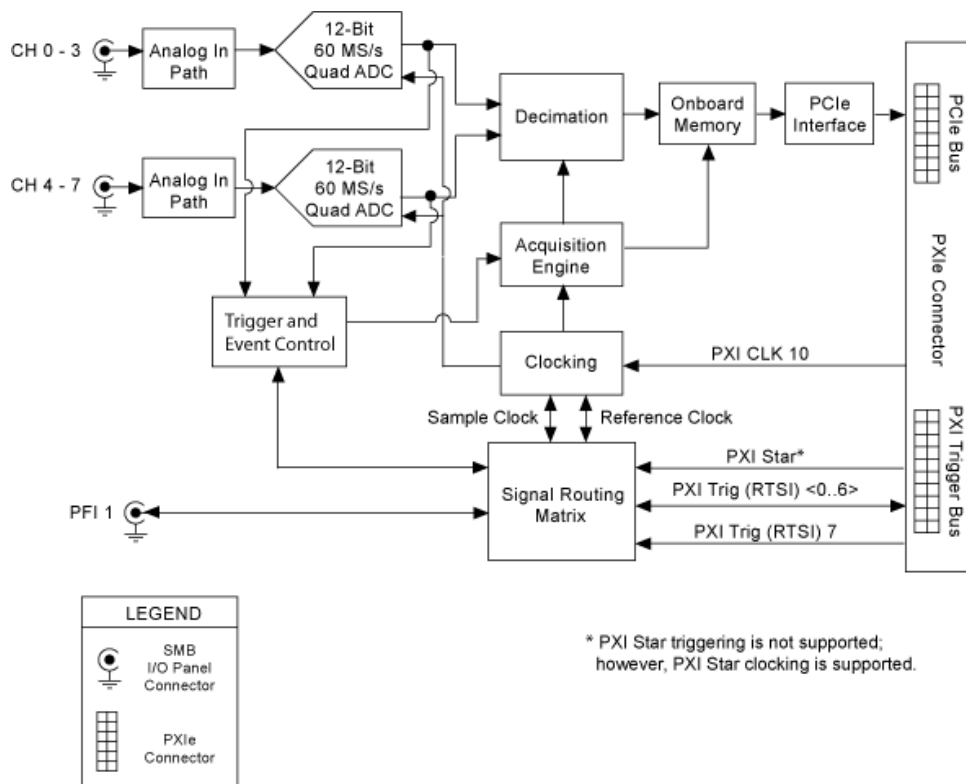
- 8 channels, simultaneously sampled
- 12-bit vertical resolution
- 60 MS/s real-time sampling rate
- 60 MHz bandwidth and antialias filter
- 16 MB, 128 MB, or 512 MB of memory
- [NI-TClk synchronization](#)
- PXI, PXIe, or PCI versions

Related Topics

[Features Supported by Device](#)

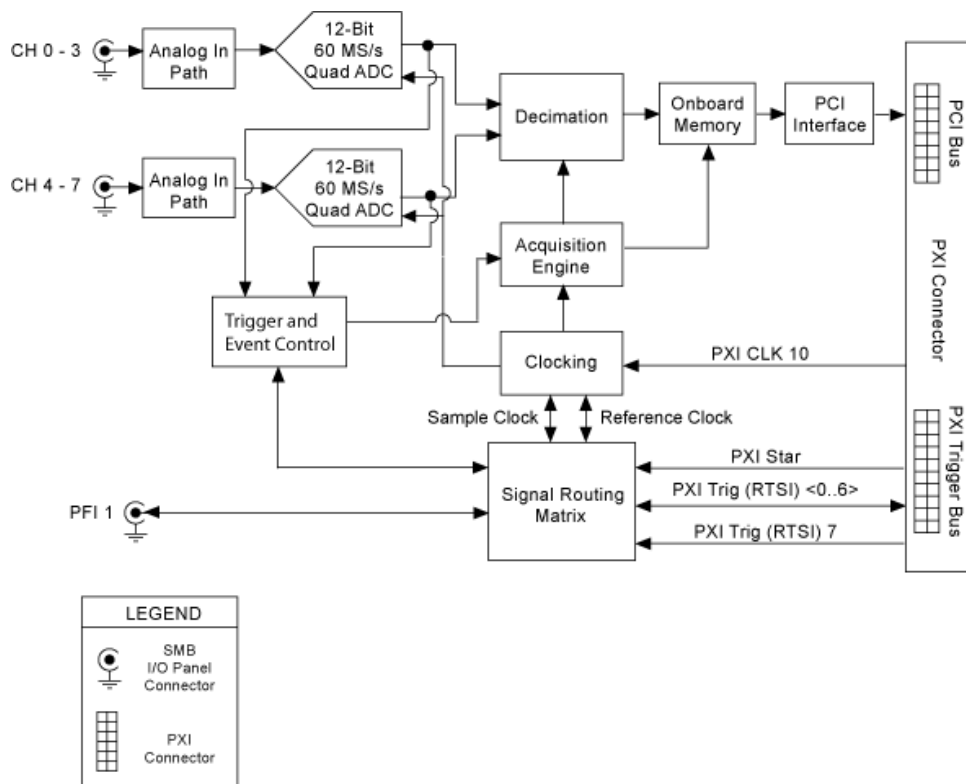
NI PXIe-5105 Block Diagram

The following figure shows a detailed block diagram of the NI PXIe-5105.



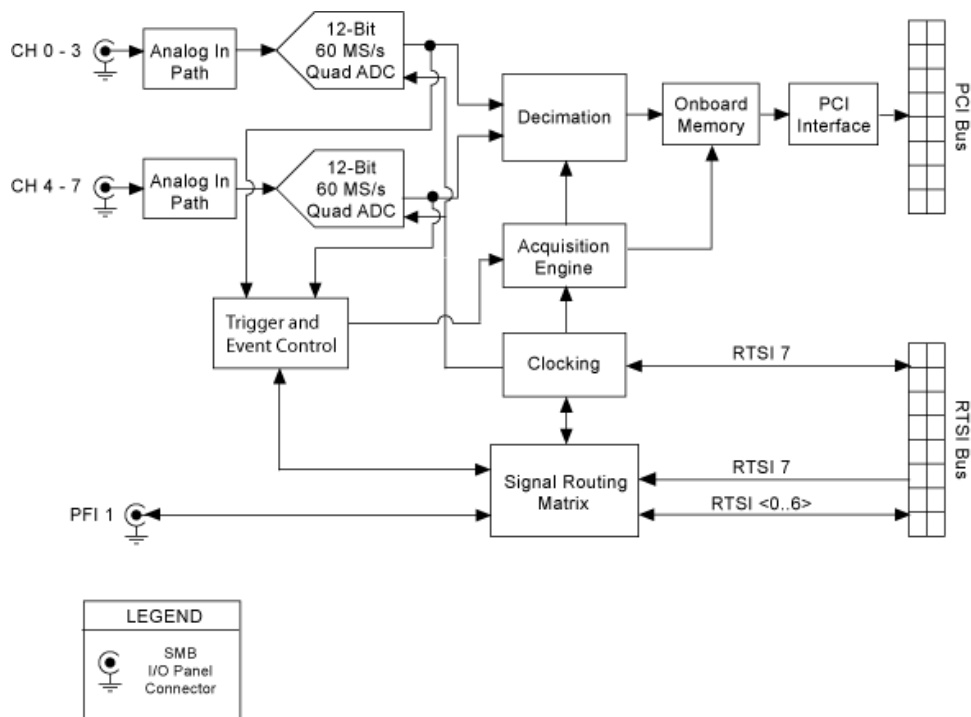
NI PXI-5105 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5105.



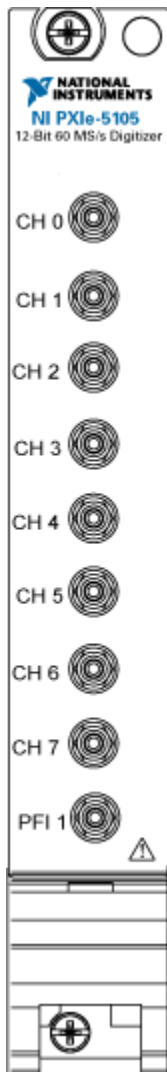
NI PCI-5105 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5105.



NI PXIe-5105 Front Panel

The following figure shows the front panel of the NI PXIe-5105. Descriptions of the connectors are shown below.



Connectors

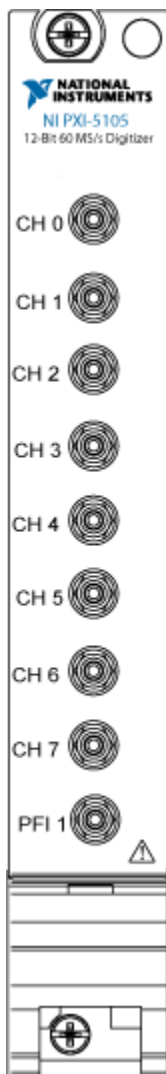
The NI 5105 has the following connectors on the front panel.

Connector	Description	Function
CH 0 through CH 7	Standard SMB connector	Analog input connection; digitizes data and triggers acquisitions
PFI 1	Standard SMB connector	Multipurpose PFI line for trigger in/out, external clock in, referen

ce clock in/out, and timebase o
ut.

NI PXI-5105 Front Panel

The following figure shows the front panel of the NI PXI-5105. Descriptions of the connectors are shown below.



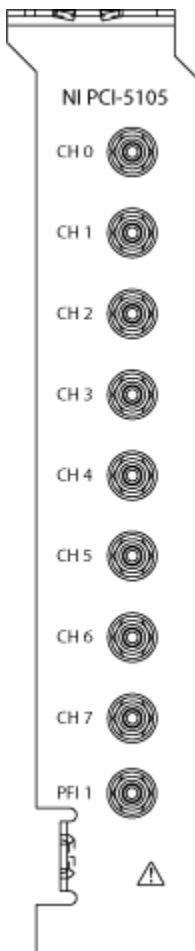
Connectors

The NI 5105 has the following connectors on the front panel.

Connector	Description	Function
CH 0 through CH 7	Standard SMB connector	Analog input connection; digitizes data and triggers acquisitions
PFI 1	Standard SMB connector	Multipurpose PFI line for trigger in/out, external clock in, reference clock in/out, and timebase out.

NI PCI-5105 Front Panel

The following figure shows the front panel of the NI PCI-5105. Descriptions of the connectors are shown below.



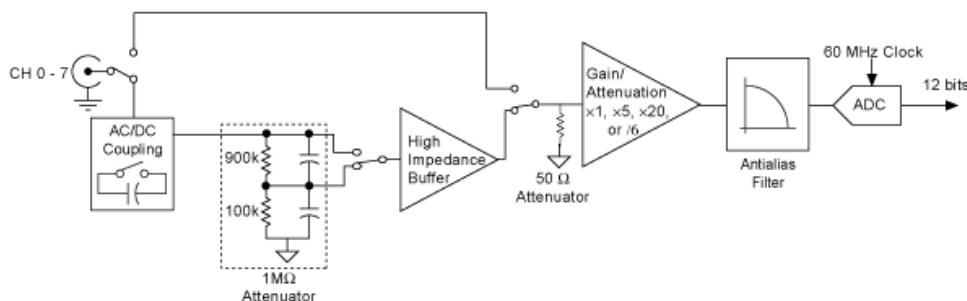
Connectors

The NI 5105 has the following connectors on the front panel.

Connector	Description	Function
CH 0 through CH 7	Standard SMB connector	Analog input connection; digitizes data and triggers acquisitions
PFI 1	Standard SMB connector	Multipurpose PFI line for trigger in/out, external clock in, reference clock in/out, and timebase output.

NI 5105 Input Signal Conditioning

The NI 5105 has eight independent input channels. Each channel provides the choice of a 50 Ω input impedance path or a 1 M Ω input impedance path, as shown in the following diagram.



The 50 Ω path is optimized for flatness, distortion, dynamic range, and noise because the active circuitry in the signal path is minimized. High-performance communications measurements can take advantage of the large [SFDR](#) and 12-bit dynamic range of the 50 Ω path.

The 1 M Ω path provides the traditional oscilloscope input impedance and loading characteristics for numerous general-purpose measurement applications.



Note The ground on the device inputs is connected to the chassis ground.

NI 5105 Input Ranges

The following table shows the NI 5105 input ranges for both the 50 Ω and 1 M Ω input paths.

50 Ω Input Path	1 M Ω Input Path
0.05 V _{pk-pk}	0.05 V _{pk-pk}
0.2 V _{pk-pk}	0.2 V _{pk-pk}
1 V _{pk-pk}	1 V _{pk-pk}

$6 V_{pk-pk}$	$6 V_{pk-pk}$
—	$30 V_{pk-pk}$

NI 5105 AC/DC Coupling

You can select AC or DC [input coupling](#) for the 1 M Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness.

The 50 Ω input path is always DC-coupled.

NI 5105 Input Impedance

You can set the NI 5105 analog input impedance to either 50 Ω or 1 M Ω . The 50 Ω path provides slightly better noise and distortion performance because the high impedance buffer for the 1 M Ω path is bypassed. The 1 M Ω path is required in applications that require minimal loading.

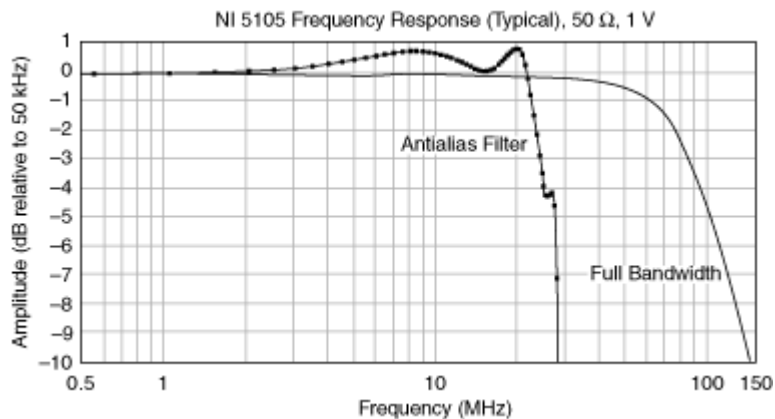
Protection

The 50 Ω inputs of the NI 5105 are protected by an overvoltage protection circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω .

NI 5105 Antialias Filter

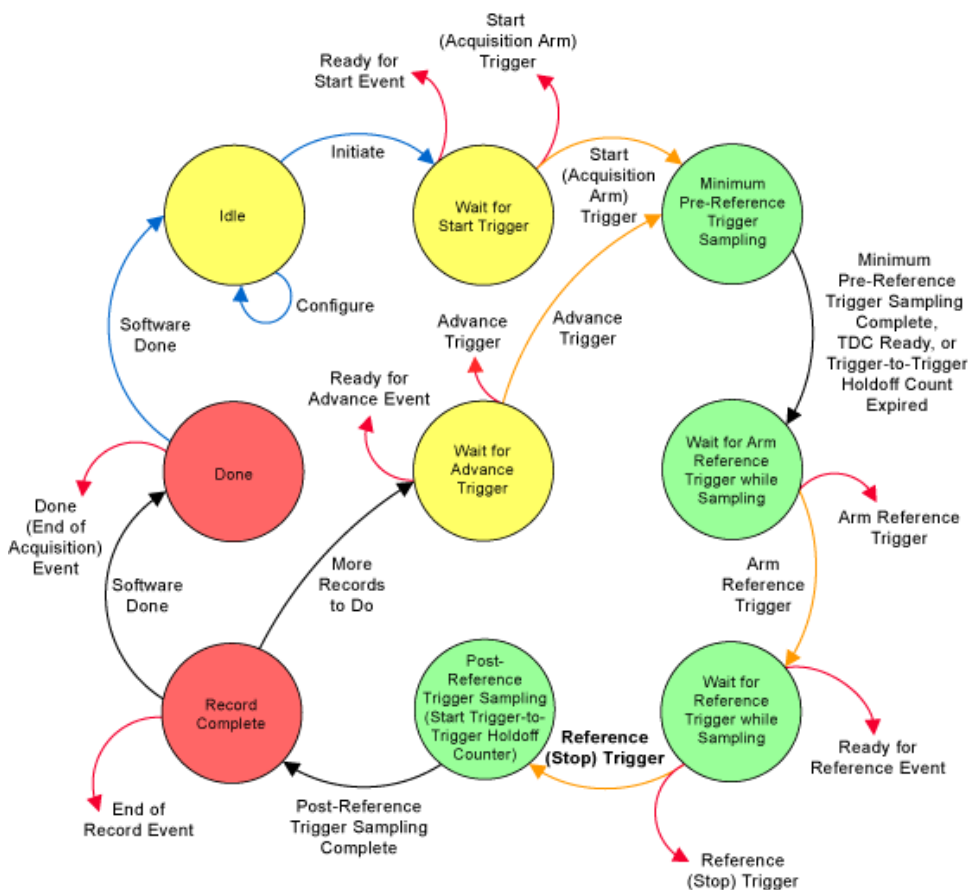
The NI 5105 provides a bandwidth-limiting filter, the 24 MHz antialias filter. The primary purpose of the antialias filter is to minimize [aliasing](#) effects caused by signals greater than [Nyquist](#) applied to the 60 MS/s ADC. Refer to the [hardware specifications document](#) for more information on this filter.

A typical response with the antialias filter is shown in the following figure.



SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.





Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the `niScope Configure Trigger` function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the

minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

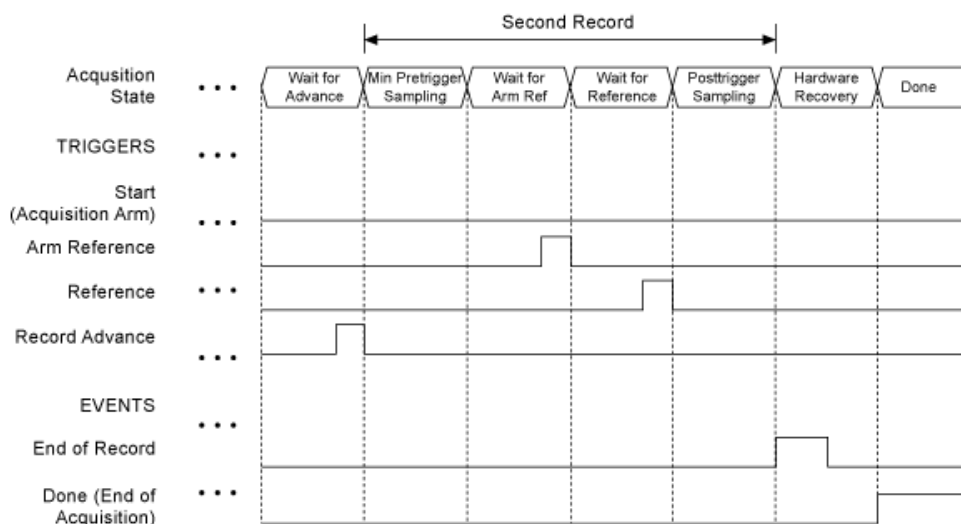
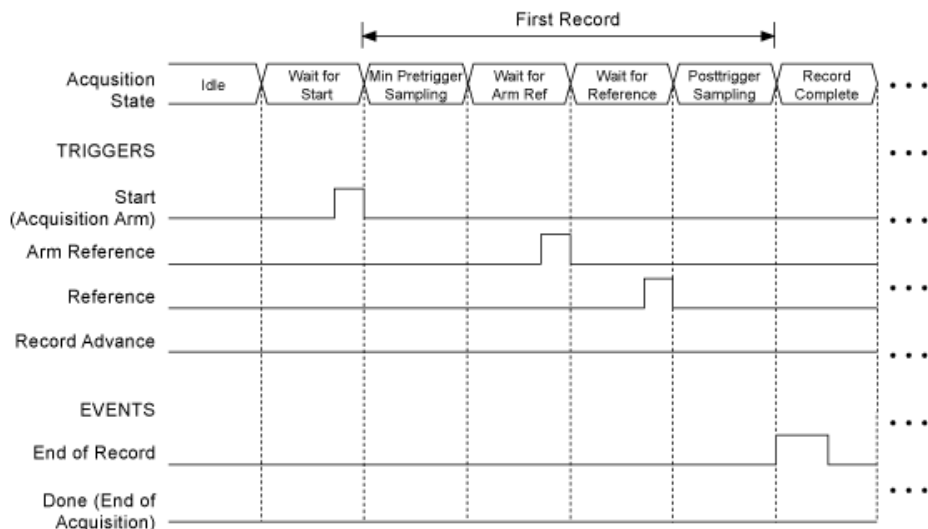
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)


SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.

 Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

PCI and PXI Ground Loop Noise

Some computer chassis can induce ground loop noise in measurements taken with instruments having grounded input connectors. Spurious DC and low-frequency content can result. The cause is usually power supply return currents from the motherboard that partially flow through the computer chassis to the power supply housing rather than entirely through the power supply cabling.

Follow these guidelines to mitigate the ground loop noise for PCI devices:

1. Ensure that the motherboard is tightly screwed down to the chassis so that chassis return currents will flow through the chassis under the motherboard and not in the vicinity of the PCI boards.
2. Ensure that the power supply cable is well seated on the motherboard.
3. Locate analog input cards in slots furthest away from the power supply.
4. Locate video cards and other high power PCI boards in slots furthest removed from the analog input cards.

Follow these guidelines to mitigate the ground loop noise for PXI devices:

1. Ensure that the screws on the front panel of the device are tightened down.
2. Ensure that any external equipment that is providing a signal, and is not floating with respect to ground (for example, it is not a passive sensor or a battery-operated device), is connected to the same ground as the chassis.
3. On rack mounted systems, ground the chassis to the metal rack mount to reduce ground loop currents.

NI PXIe-5105 Routing Matrix

The following table shows the signals available for export from the NI PXIe-5105, and the lines to which they can be routed.

Source	Destination
--------	-------------

	PXI_Trig <0..6> (PXI Bus)	PFI 1
Exported Clocks		
Reference Clock (External)	—	√
Sample Clock (Full Rate)	—	√
Triggers		
Acquisition Arm (Start) Trigger	√	√
Reference (Stop) Trigger	√	√
Events		
End of Record Event	√	√
End of Acquisition Event	√	√
Ready for Start Event	√	√
Ready for Reference Event	√	√
Ready for Advance Event	√	√

NI PXI-5105 Routing Matrix

The following table shows the signals available for export from the NI PXI-5105, and the lines to which they can be routed.

Source	Destination	
	PXI_Trig <0..6> (PXI Bus)	PFI 1
Exported Clocks		
Reference Clock (External)	—	√
Sample Clock (Full Rate)	—	√
Triggers		
Acquisition Arm (Start) Trigger	√	√
Reference (Stop) Trigger	√	√
Events		
End of Record Event	√	√
End of Acquisition Event	√	√
Ready for Start Event	√	√
Ready for Reference Event	√	√

Ready for Advance Event	√	√
-------------------------	---	---

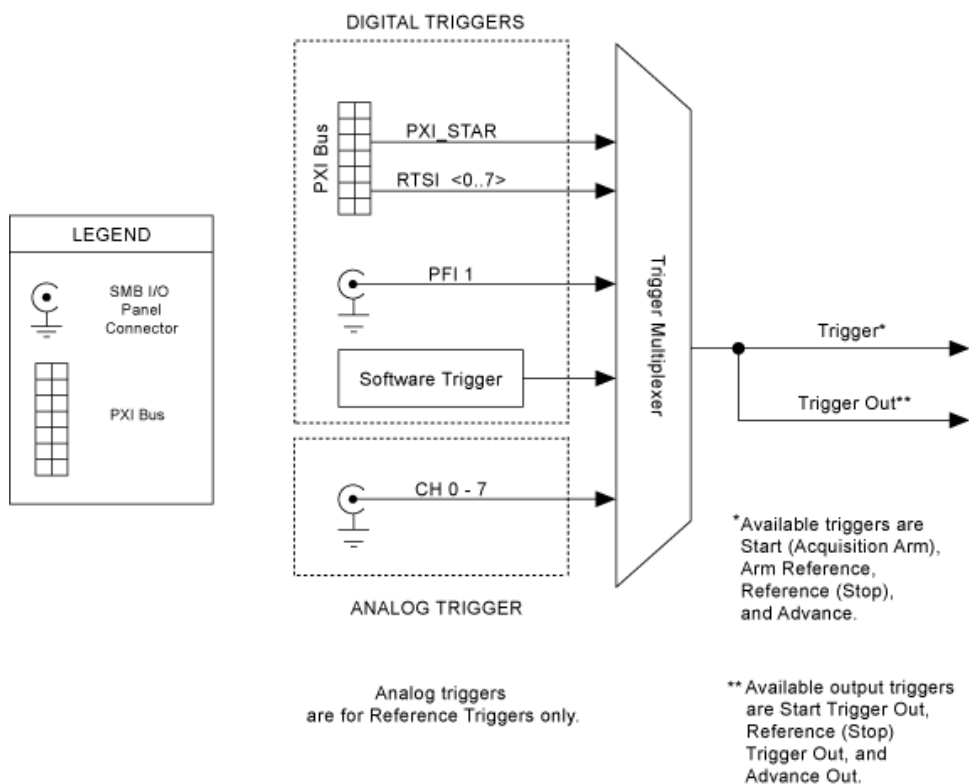
NI PCI-5105 Routing Matrix

The following table shows the signals available for export from the NI PCI-5105, and the lines to which they can be routed.

Source	Destination		
	RTSI <0..6> (RTSI Bus)	PFI 1	RTSI 7
Exported Clocks			
Reference Clock (External)	—	√	—
Reference Clock (Internal)	√	√	√
Sample Clock (Full Rate)	—	√	—
Triggers			
Acquisition Arm (Start) Trigger	√	√	—
Reference (Stop) Trigger	√	√	—
Events			
End of Record Event	√	√	—
End of Acquisition Event	√	√	—
Ready for Start Event	√	√	—
Ready for Reference Event	√	√	—
Ready for Advance Event	√	√	—

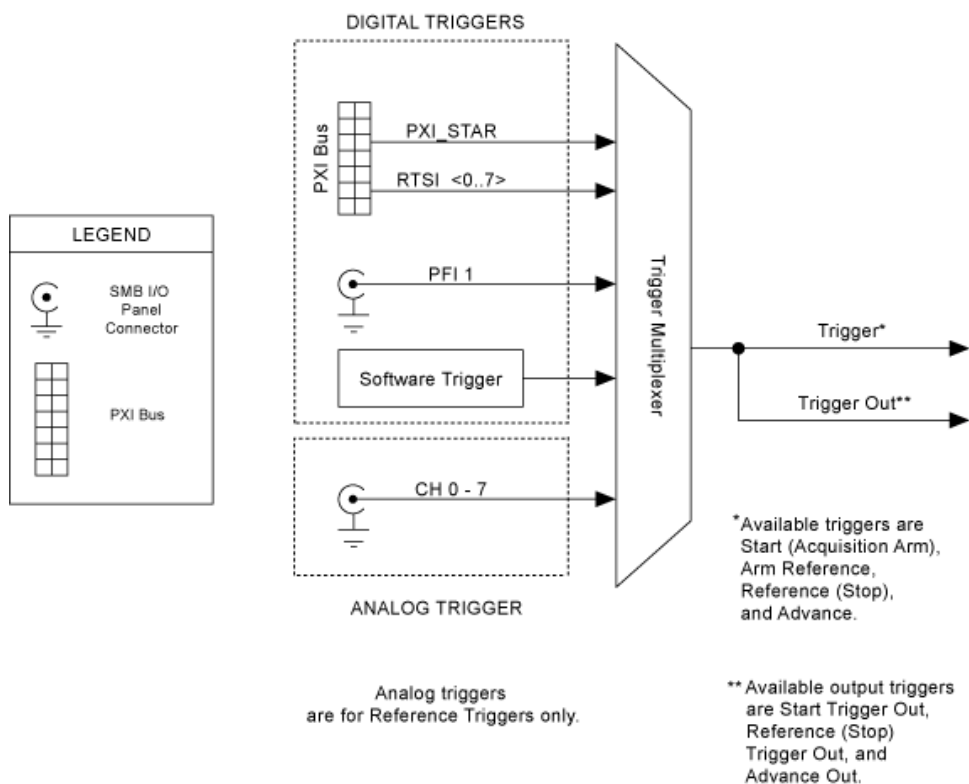
NI PXIe-5105 Trigger Sources

The following figure shows the trigger sources for the NI PXIe-5105.



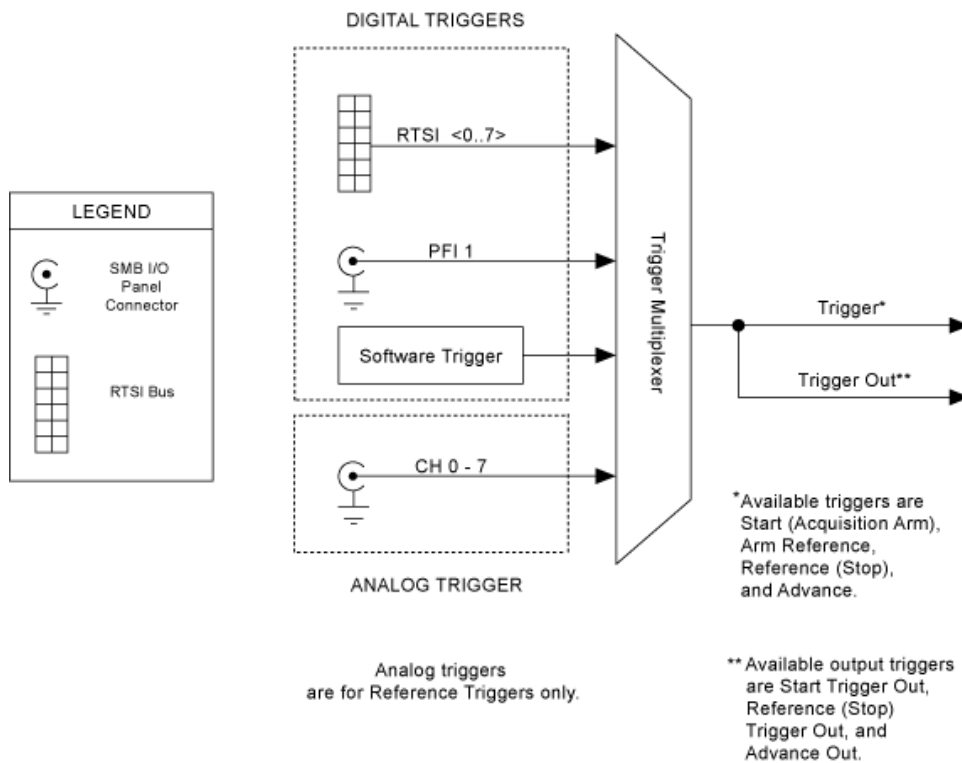
NI PXI-5105 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5105.



NI PCI-5105 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5105.



NI 5105 Trigger Types

The NI 5105 supports the following trigger types: [edge](#), [hysteresis](#), [immediate](#), [digital](#), [software](#), and [window](#) triggers.

The NI 5105 uses the predecimated digitized data from the input channel ADC when performing an analog trigger on any of the channels. This behavior ensures a level and time trigger accuracy of 1 timebase clock period. Because the trigger detector circuit is after the ADC, there is a 24- to 30-sample delay when you export an analog trigger to another device.

NI 5105 Trigger Holdoff

For the NI 5105, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its

Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

NI 5105 Trigger Delay

Trigger delay, which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

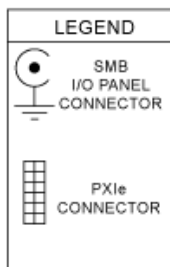
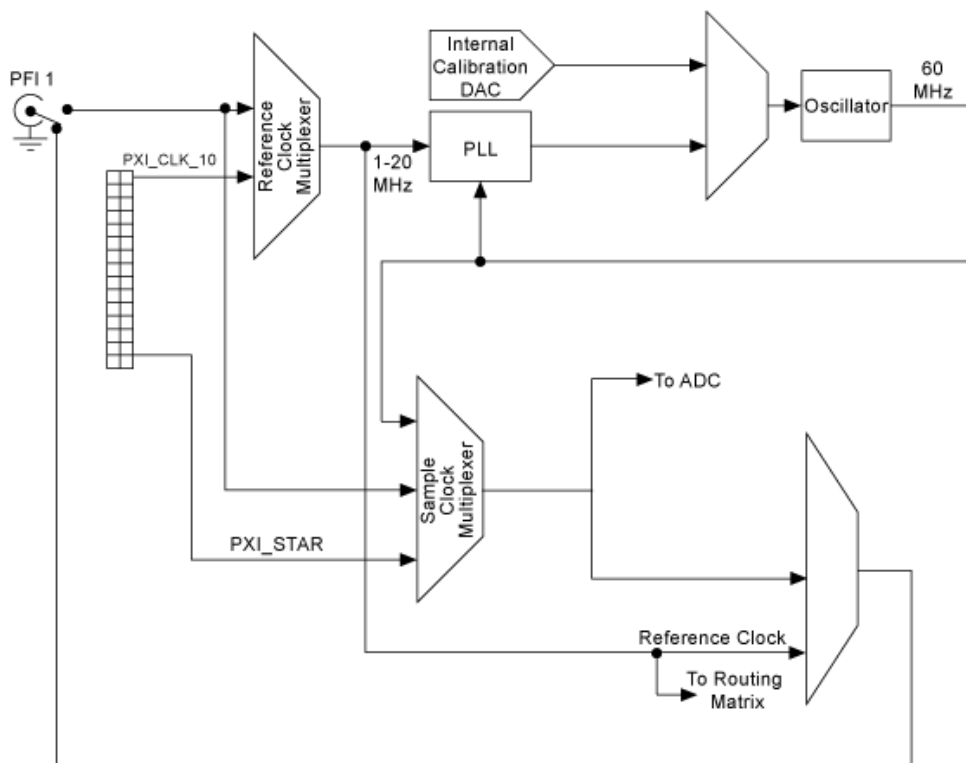
$$\text{Max trigger delay in seconds} = [(2^{32} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock.

PXIe-5105 Clocking

The clock circuitry on the PXIe-5105 offers versatile clocking options with its ability to use either the internal 60 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the PXIe-5105 to phase lock the internal 60 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the PXIe-5105.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The PXIe-5105 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The PXIe-5105 has an onboard voltage controlled crystal oscillator (VCXO) running at 60 MHz. When using the onboard 60 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 60 MHz

frequency of the VCXO. In PLL mode, the device [phase locks](#) its 60 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the PXIe-5105 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 60 MHz clock. In these cases, the PXIe-5105 can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the hardware [specifications document](#) for external sample clock requirements.

Reference Clock

The [reference clock](#) is used in the PXIe-5105 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The PXIe-5105 can accept a reference clock from its front panel PFI 1 connector as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to PFI 1. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the PXIe-5105 reference clock is `None`, or no reference clock is used.



Note Locking to a reference clock is not valid when using an external sample clock.

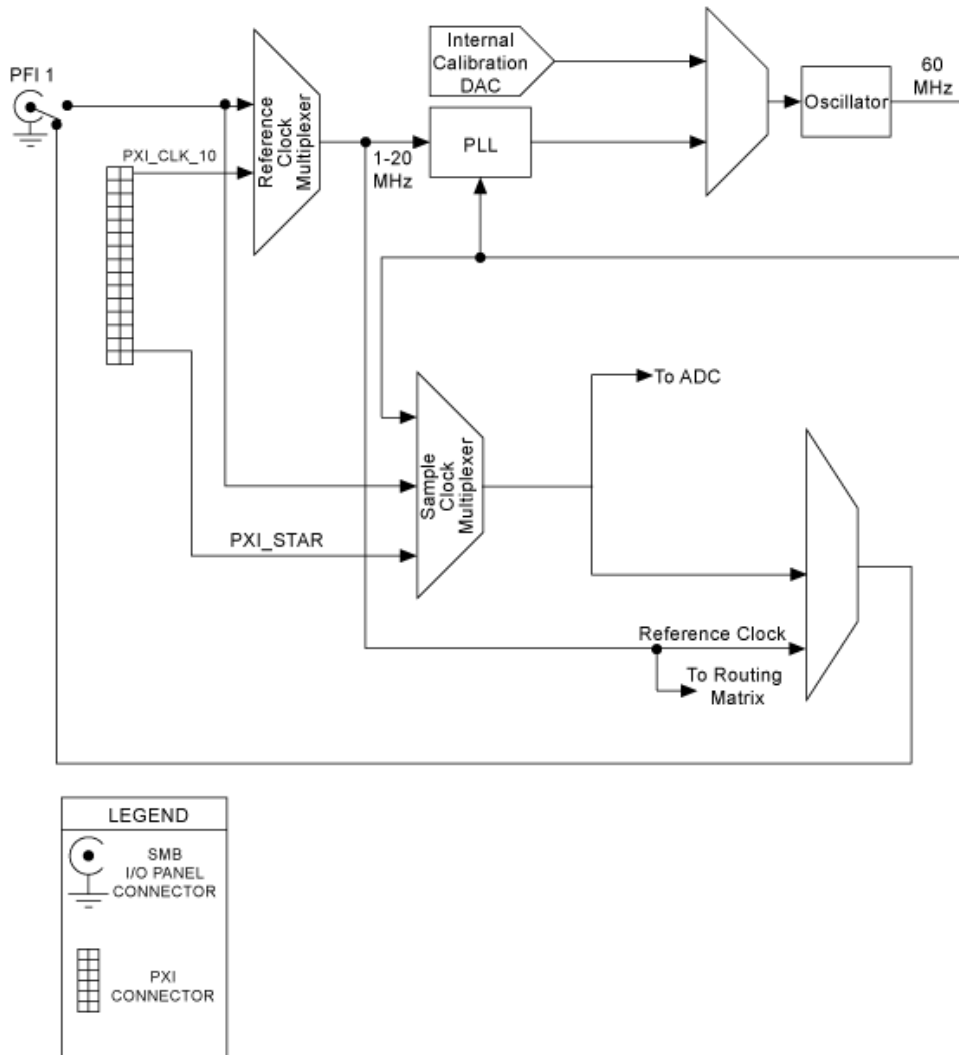
Exporting Clocks

The PFI 1 connector on the PXIe-5105 can be used for exporting the 60 MHz internal sample clock (full rate) or the reference clock when phase-locked to PXI_CLK10.

PXI-5105 Clocking

The clock circuitry on the PXI-5105 offers versatile clocking options with its ability to use either the internal 60 MHz sample clock or to accept an external sample clock

that you provide. You can also use the phase-locked loop (PLL) circuit on the PXI-5105 to phase lock the internal 60 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the PXI-5105.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The PXI-5105 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The PXI-5105 has an onboard voltage controlled crystal oscillator (VCXO) running at 60 MHz. When using the onboard 60 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 60 MHz frequency of the VCXO. In PLL mode, the device [phase locks](#) its 60 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the PXI-5105 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 60 MHz clock. In these cases the PXI-5105 can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the hardware [specifications document](#) for external sample clock requirements.

[Reference Clock](#)

The [reference clock](#) is used in the PXI-5105 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The PXI-5105 can accept a reference clock from its front panel PFI 1 connector as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to PFI 1. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the PXI-5105 reference clock is `None`, or not to use a reference clock.



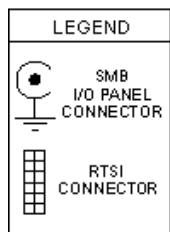
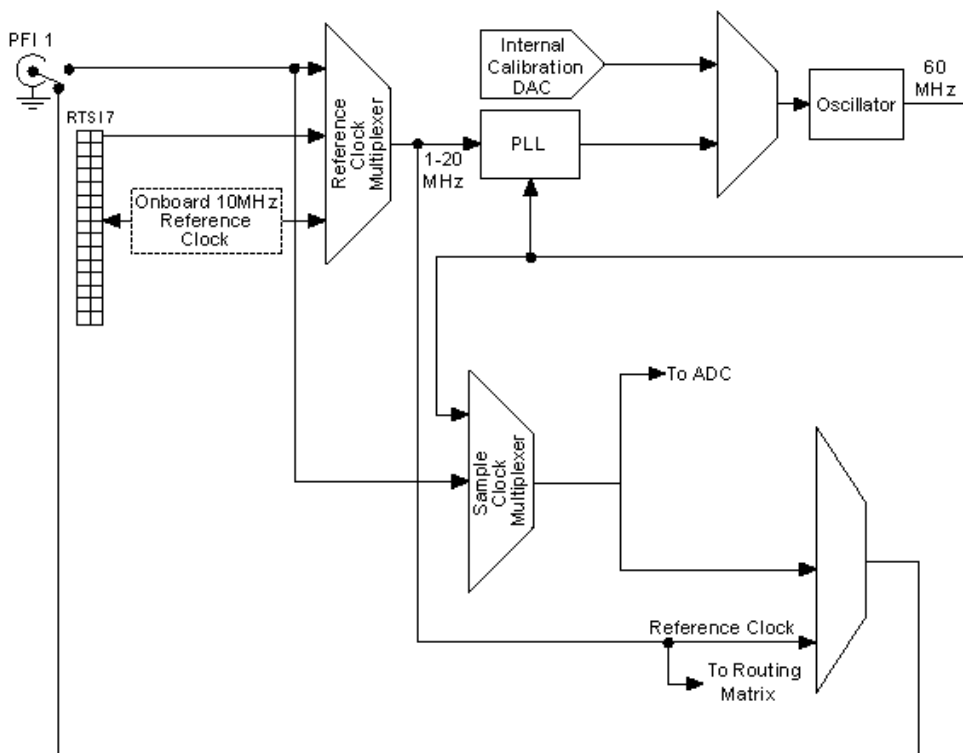
Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Clocks

The PFI 1 connector on the PXI-5105 can be used for exporting the 60 MHz internal sample clock (full rate) or the reference clock when phase-locked to PXI_CLK10.

PCI-5105 Clocking

The clock circuitry on the PCI-5105 offers versatile clocking options with its ability to use either the internal 60 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the PCI-5105 to phase lock the internal 60 MHz sample clock with an external reference clock that you provide. The following diagram shows the clocking options of the PCI-5105.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The PCI-5105 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The PCI-5105 has an onboard voltage controlled crystal oscillator (VCXO) running at 60 MHz. When using the onboard 60 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 60 MHz frequency of the VCXO. In PLL mode, the device [phase locks](#) its 60 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the PCI-5105 with other devices in a measurement system.


External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 60 MHz clock. In these cases the PCI-5105 can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector. Refer to the hardware [specifications document](#) for external sample clock requirements.

[Reference Clock](#)

The [reference clock](#) is used in the PCI-5105 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The PCI-5105 can accept a reference clock from its front panel PFI 1 connector as well as from RTSI 7 (RTSI Clock) or the internal 10 MHz reference clock. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to PFI 1. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase

locked to the same reference clock. The default setting for the PCI-5105 reference clock is `None`, or not to use a reference clock.

 **Note** Locking to a reference clock is not valid when using external sample clocking.

Exporting Clocks

The PFI 1 connector on the PCI-5105 can be used for exporting the 60 MHz internal sample clock (full rate) or the reference clock.

NI 5105 Onboard Memory

The NI 5105 allocates at least 256 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is approximately 128 12-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is determined by the the size of the memory option divided by 256 bytes (128 samples), as shown in the following equation:

$$\frac{[(\text{Record length}^* \times 2^\dagger \times \text{number of enabled channels}) + 480^\ddagger] \text{ rounded up to nearest 128 bytes}}{256} = \text{Number of records}$$

Note: The maximum number of records is 100,000.

*samples
 †bytes/sample
 ‡bytes

The memory options are 16 MB, 128 MB, or 512 MB, and the available memory is divided among all enabled channels.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5105 continues to acquire posttrigger samples if you have specified a posttrigger sample

count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

NI 5105 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn

more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5105, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges and all filter paths (enabled/disabled).
- Calibrates trigger timing to ensure accurate trigger timing and time-stamping.
- Calibrate the digital clock manager (DCM)
- Takes approximately 50 seconds to complete.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5105 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature

variations are a serious problem, self-calibration is not recommended more than once per day.

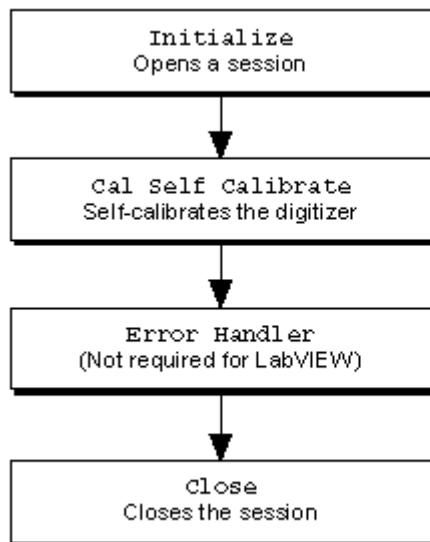
Input Connections During Self-Calibration

The NI 5105 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at **Start»All Programs»National Instruments»NI-SCOPE»Examples**.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger timing	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

NI 5114

NI 5114 high-speed digitizers feature deep onboard acquisition memory and SDTV, EDTV, and HDTV triggering. This digitizer meets the needs of test systems that make automated time-domain measurements.

Features

The NI 5114 has the following features:

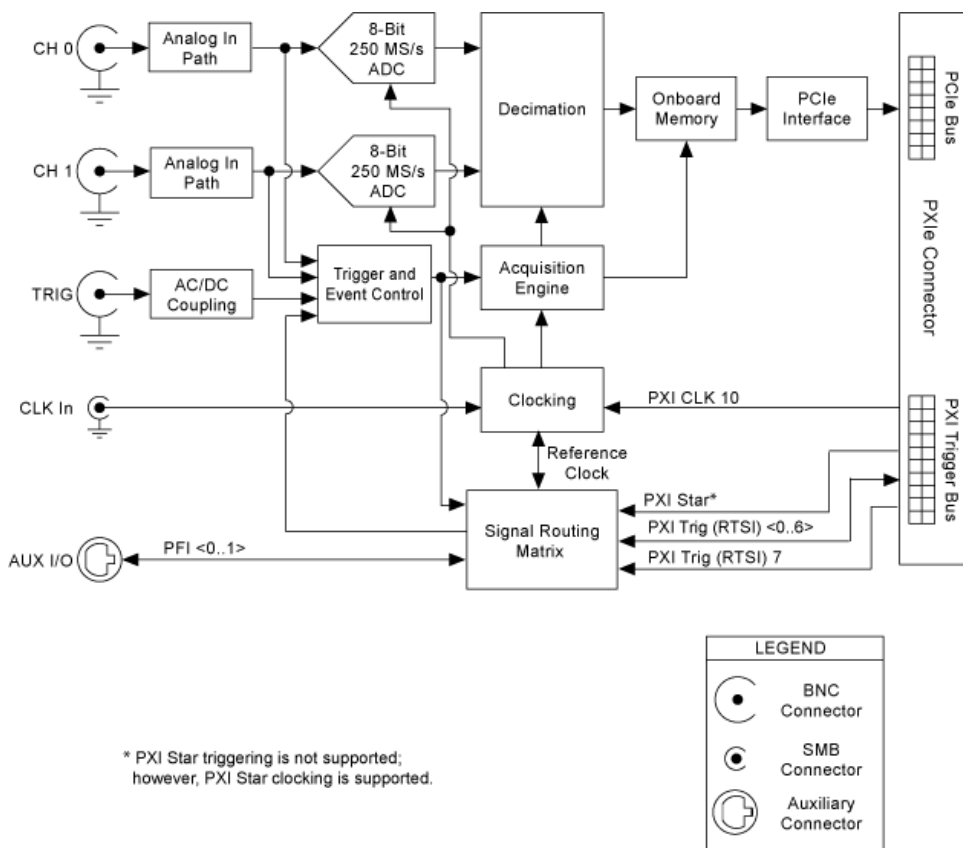
- 2 channels
- 8-bit vertical resolution
- 250 MS/s real-time sampling rate
- 5.0 GS/s random interleaved sampling (RIS) rate
- 125 MHz bandwidth
- 8, 64, or 256 MB of memory per channel
- 1 MΩ or 50 Ω selectable input impedance
- AC/DC/GND coupling
- Analog, digital, SDTV/EDTV/HDTV video triggering
- PXI, PXIe, or PCI versions
- [NI-TClk synchronization](#)

Related Topics

Features Supported by Device

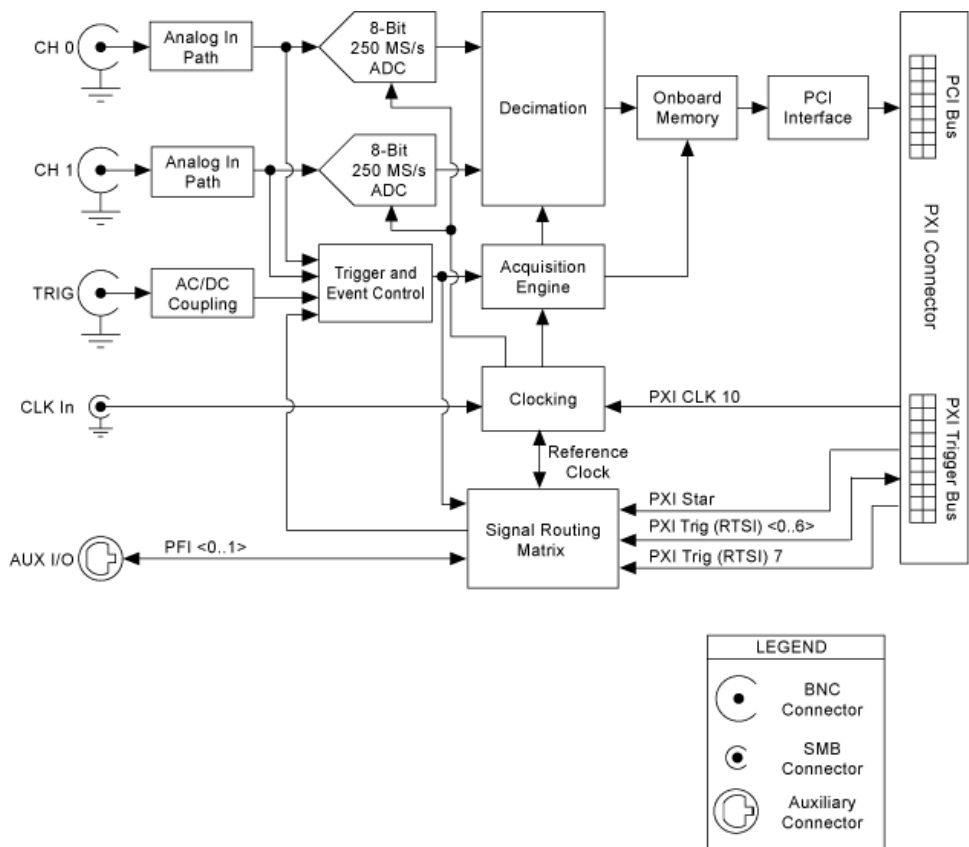
NI PXIe-5114 Block Diagram

The following figure shows a detailed block diagram of the NI PXIe-5114.



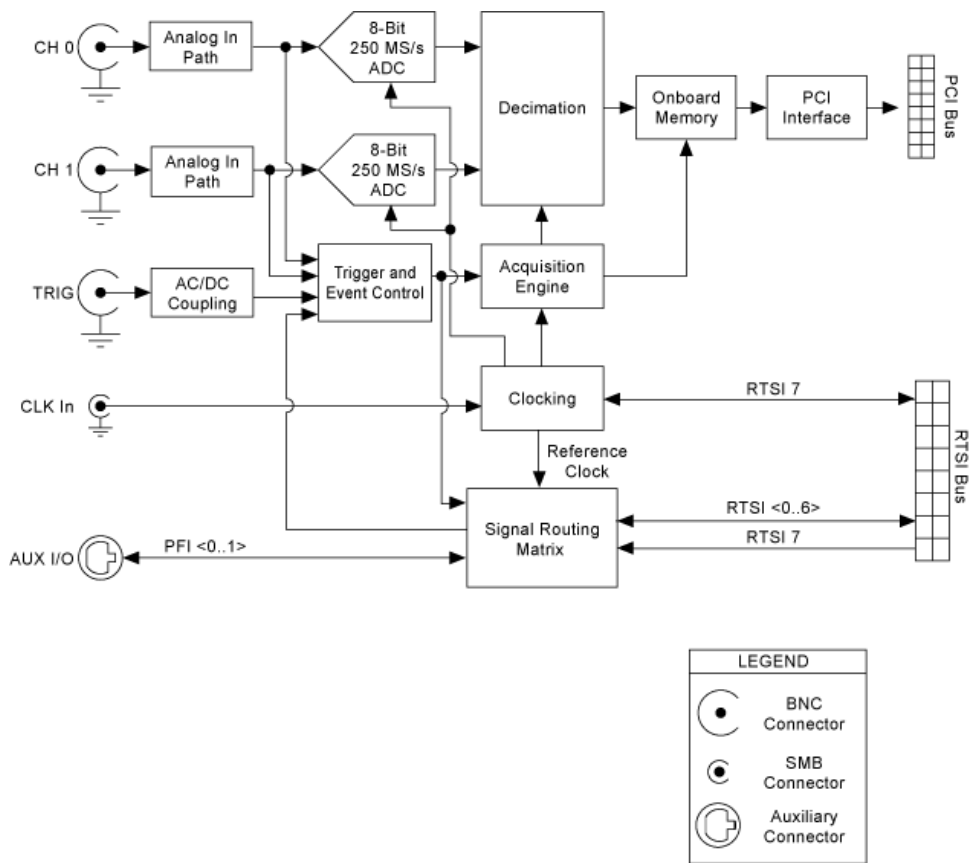
NI PXI-5114 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5114.



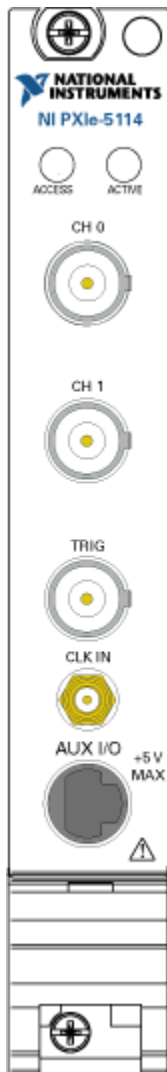
NI PCI-5114 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5114.



NI PXIe-5114 Front Panel

The following figure shows the front panel of the NI PXIe-5114. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI PXIe-5114 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

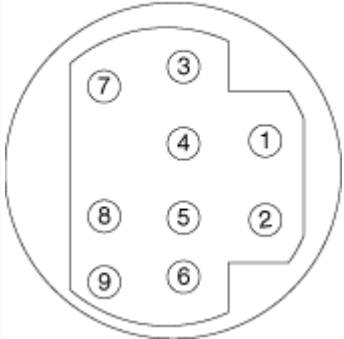
Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—The module is unable to detect the external sample clock. ▪ Overvoltage error—The module has detected a 50 Ω overvoltage error. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>


Connectors

The NI 5114 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

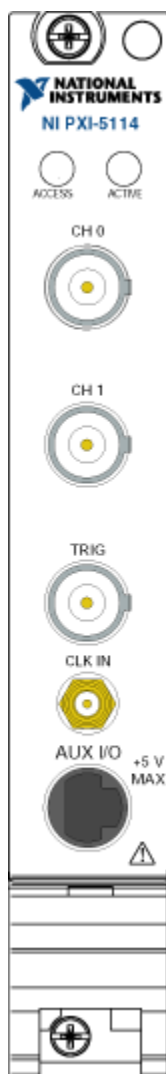
AUX I/O Connector Pin Assignments

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PXI-5114 Front Panel

The following figure shows the front panel of the NI PXI-5114. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI PXI-5114 has two LEDs to indicate status: Access and Active.


Access LED

The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.
Green	The module is ready to be programmed by NI-SCOPE.

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—The module is unable to detect the external sample clock. ▪ Overvoltage error—The module has detected a 50 Ω overvoltage error. <p> Note Certain driver interactions may cause the Active LED to flash red. An error cond</p>

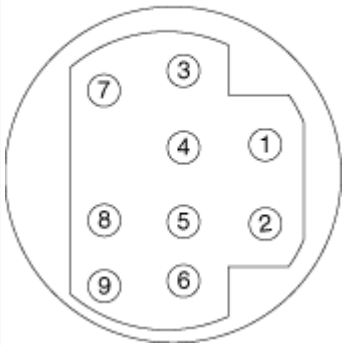
ition does not exist unless the Active LED r
emains red.

Connectors


The NI 5114 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

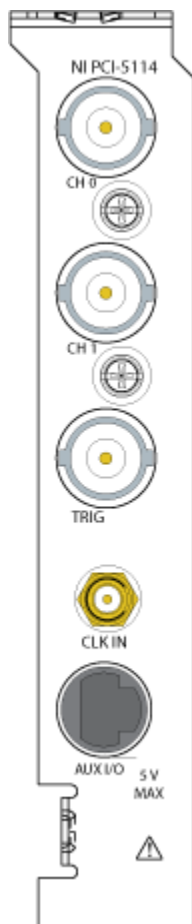
9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1

	7	Reserved
	8	Reserved
	9	PFI0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PCI-5114 Front Panel

The following figure shows the front panel of the NI PCI-5114. Descriptions of the connectors are shown below.



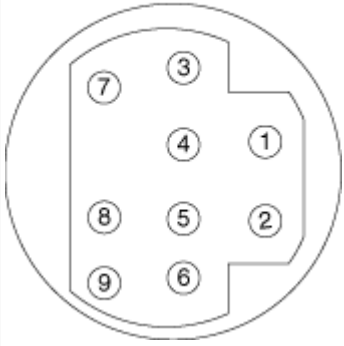
Connectors

The NI PCI-5114 has the following connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

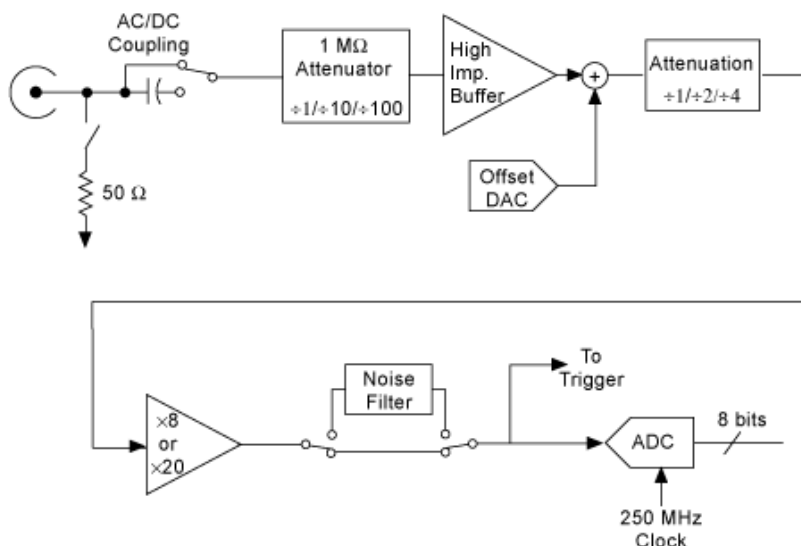
PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

Note Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI 5114 Input Signal Conditioning

The NI 5114 provides two independent digitizer input channel signal conditioning paths. Each path provides you with a choice of 50 Ω input impedance or 1 MΩ input impedance, as shown in the following diagram.



Note The ground on the device inputs is connected to the chassis ground.

NI 5114 Input Ranges

As shown in the following table, the 1 MΩ path supports up to 40 V_{pk-pk} whereas the 50 Ω path supports up to 10 V_{pk-pk}.

50Ω Input Path	1MΩ Input Path
0.04 V _{pk-pk}	0.04 V _{pk-pk}
0.1 V _{pk-pk}	0.1 V _{pk-pk}
0.2 V _{pk-pk}	0.2 V _{pk-pk}

$0.4 V_{pk-pk}$	$0.4 V_{pk-pk}$
$1 V_{pk-pk}$	$1 V_{pk-pk}$
$2 V_{pk-pk}$	$2 V_{pk-pk}$
$4 V_{pk-pk}$	$4 V_{pk-pk}$
$10 V_{pk-pk}$	$10 V_{pk-pk}$
—	$20 V_{pk-pk}$
—	$40 V_{pk-pk}$

NI 5114 Input Impedance

You can set the NI 5114 analog input impedance to either 50Ω or $1 M\Omega$. The $1 M\Omega$ path is required in applications that require minimal loading or that require using a standard 10:1 oscilloscope probe.

Protection

The 50Ω inputs of the NI 5114 are protected by a thermal disconnect circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important to observe the maximum signal input, especially when the inputs are set for 50Ω .

NI 5114/5122/5124/5142 AC/DC/GND Coupling


You can select AC, DC, or GND [input coupling](#) for the $1 M\Omega$ input path, and DC or GND input coupling for the 50Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit, or if you are using the 50Ω path.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5114 Vertical Offset

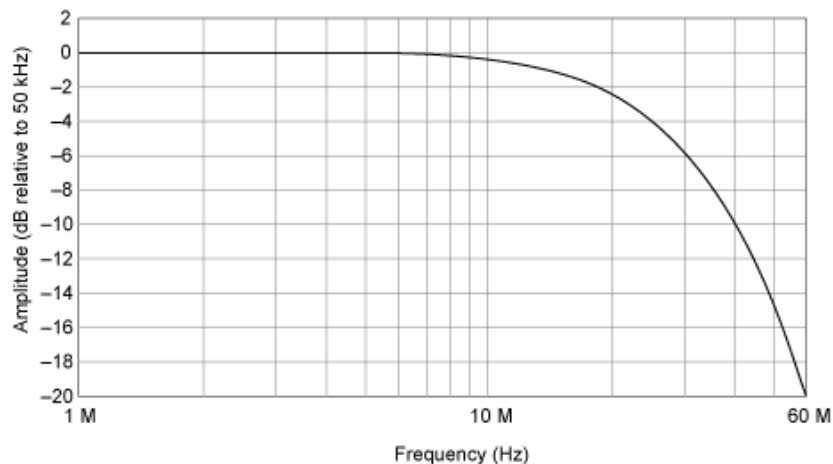
The following table shows the valid vertical offset for each range on the NI 5114.

Range	50 Ω Vertical Offset	1 M Ω Vertical Offset
0.04 V _{pk-pk}	± 0.8 V	± 0.8 V
0.1 V _{pk-pk}	± 0.8 V	± 0.8 V
0.2 V _{pk-pk}	± 0.8 V	± 0.8 V
0.4 V _{pk-pk}	± 0.8 V	± 0.8 V
1 V _{pk-pk}	± 6.5 V	± 8 V
2 V _{pk-pk}	± 6 V	± 8 V
4 V _{pk-pk}	± 5 V	± 8 V
10 V _{pk-pk}	± 2 V	± 30 V
20 V _{pk-pk}	N/A	± 25 V
40 V _{pk-pk}	N/A	± 15 V

 Note The maximum allowable DC voltage is 7 V in the 50 Ω path and 35 V in the 1 M Ω path. The vertical offsets were chosen to stay within these voltage limits.

NI 5114 Noise Filter

The NI 5114 has a 20 MHz noise filter that limits the bandwidth of the signal path through both the 1 M Ω and 50 Ω signal paths. This filter is intended to reduce noise when the signal content is 20 MHz or less. A typical frequency response of the noise filter is shown in the following figure.



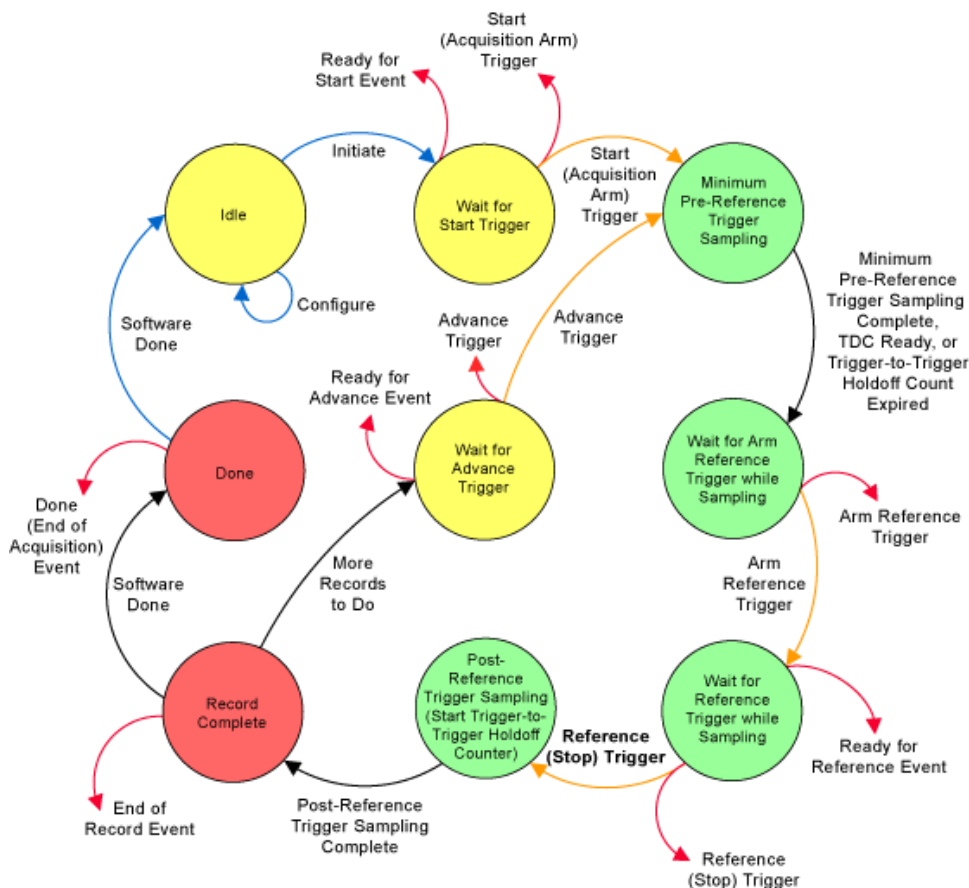
The noise filter provides over 20 dB of attenuation at 100 MHz, which is useful for rejecting out of band or background noise in lower frequency applications. An example application would be characterization of video waveforms in the presence of IF noise. Because most of the signal content is below the 8 MHz range, the 20 MHz noise filter is ideal. Many standard noise tests specify a 20 MHz noise bandwidth.



Note To enable/disable the noise filter, call the [niScope Configure Chan Characteristics VI](#) or the [niScope_ConfigureChanCharacteristics](#) function and set the Max Input Frequency parameter to the appropriate value.

SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

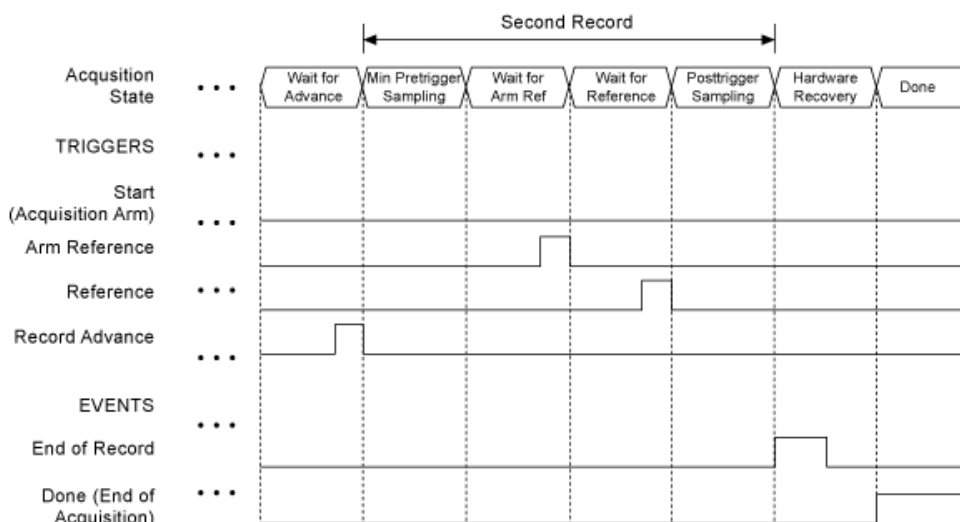
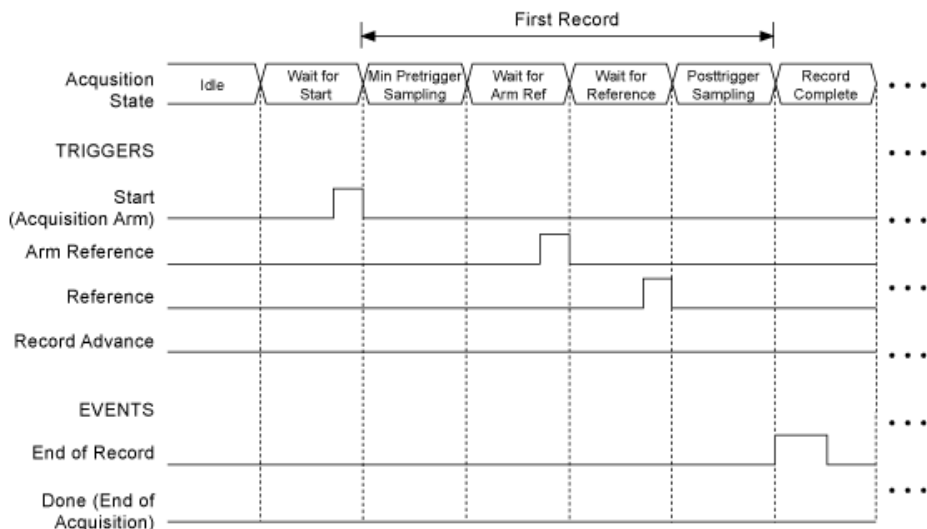
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXIe-5114 Routing Matrix

The following table shows the signals available for export from the NI PXIe-5114 and the lines to which they can be routed.

Source	Destination	
	PXI_Trig <0..6> (PXI Bus)	PFI <0..1> (AUX I/O)
Exported Clocks		
Reference Clock (External)	√	√
Triggers		
Acquisition Arm (Start) Trigger	√	√
Reference (Stop) Trigger	√	√
Events		
End of Record Event	√	√
End of Acquisition Event	√	√
Ready for Start Event	√	√
Ready for Reference Event	√	√
Ready for Advance Event	√	√

NI PXI-5114 Routing Matrix

The following table shows the signals available for export from the NI PXI-5114 and the lines to which they can be routed.

Source	Destination	
	PXI_Trig <0..6> (PXI Bus)	PFI <0..1> (AUX I/O)
Exported Clocks		
Reference Clock (External)	√	√
Triggers		
Acquisition Arm (Start) Trigger	√	√
Reference (Stop) Trigger	√	√

Events			
End of Record Event		√	√
End of Acquisition Event		√	√
Ready for Start Event		√	√
Ready for Reference Event		√	√
Ready for Advance Event		√	√

NI PCI-5114 Routing Matrix

The following table shows the signals available for export from the NI PCI-5114 and the lines to which they can be routed.

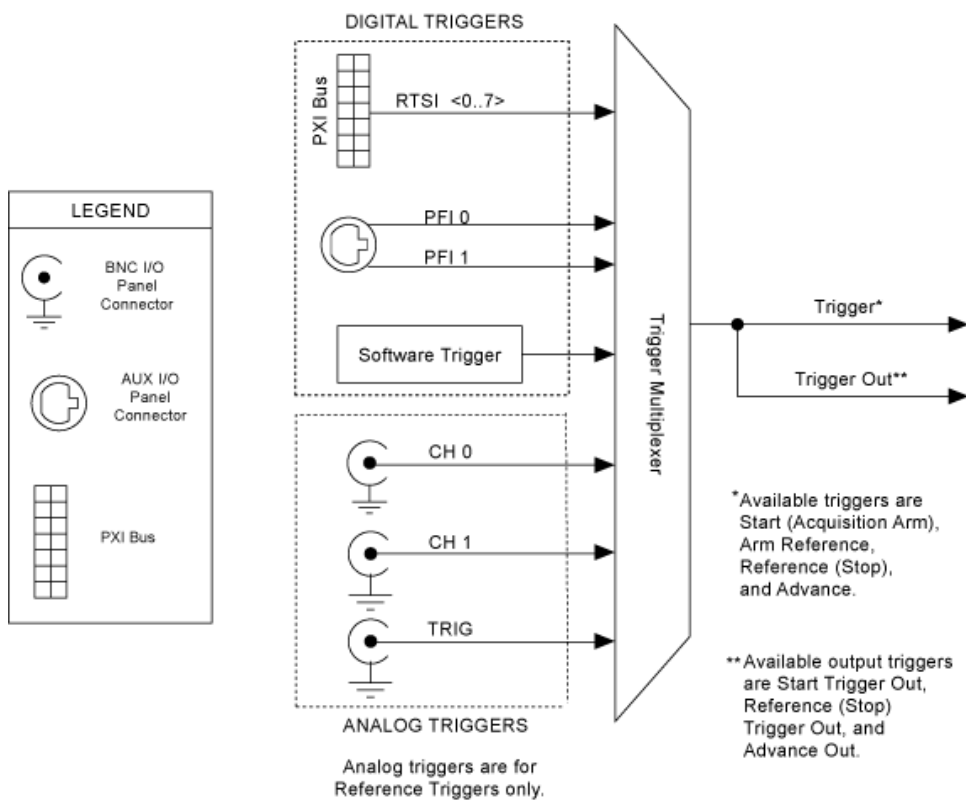
Source	Destination		
	RTSI <0.6> (RTSI Bus)	PFI <0.1> (AUX I/O)	RTSI 7
Exported Clocks			
Reference Clock (External)	—	√	—
Reference Clock (Internal)	—	√	√
Triggers			
Acquisition Arm (Start) Trigger	√	√	—
Reference (Stop) Trigger	√	√	—
Events			
End of Record Event	√	√	—
End of Acquisition Event	√	√	—
Ready for Start Event	√	√	—
Ready for Reference Event	√	√	—
Ready for Advance Event	√	√	—

NI 5114/5122/5124/5142 Analog Trigger Types

The NI 5114/5122/5124/5142 supports the following analog trigger types: edge, hysteresis, window, and video triggers.

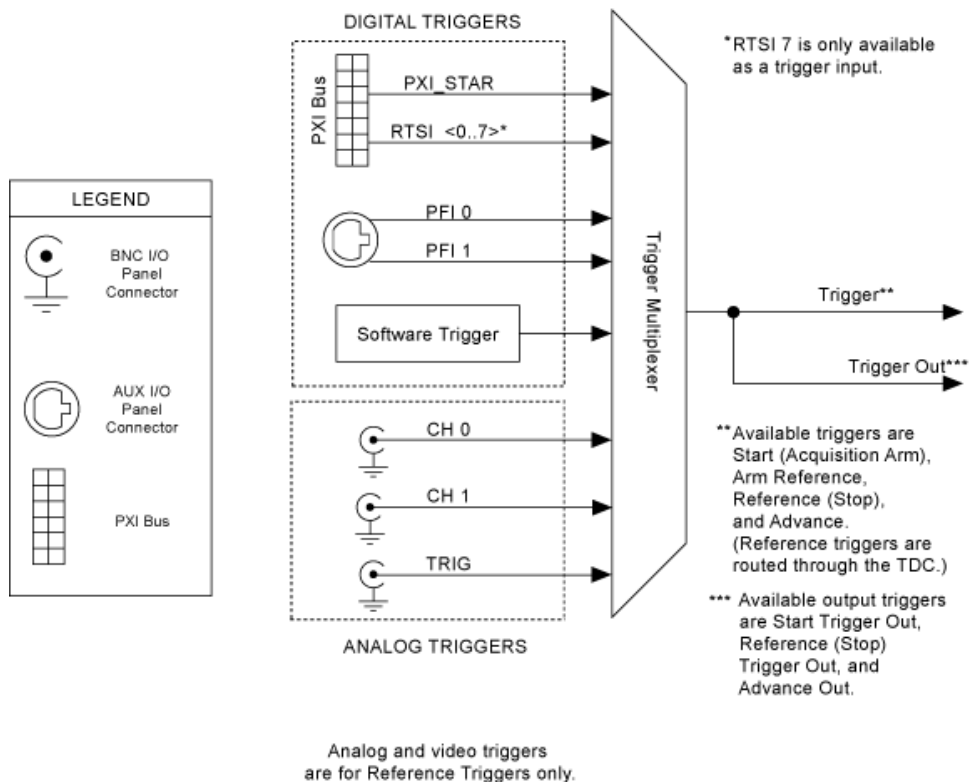
NI PXIe-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PXIe-5114/5122/5124.



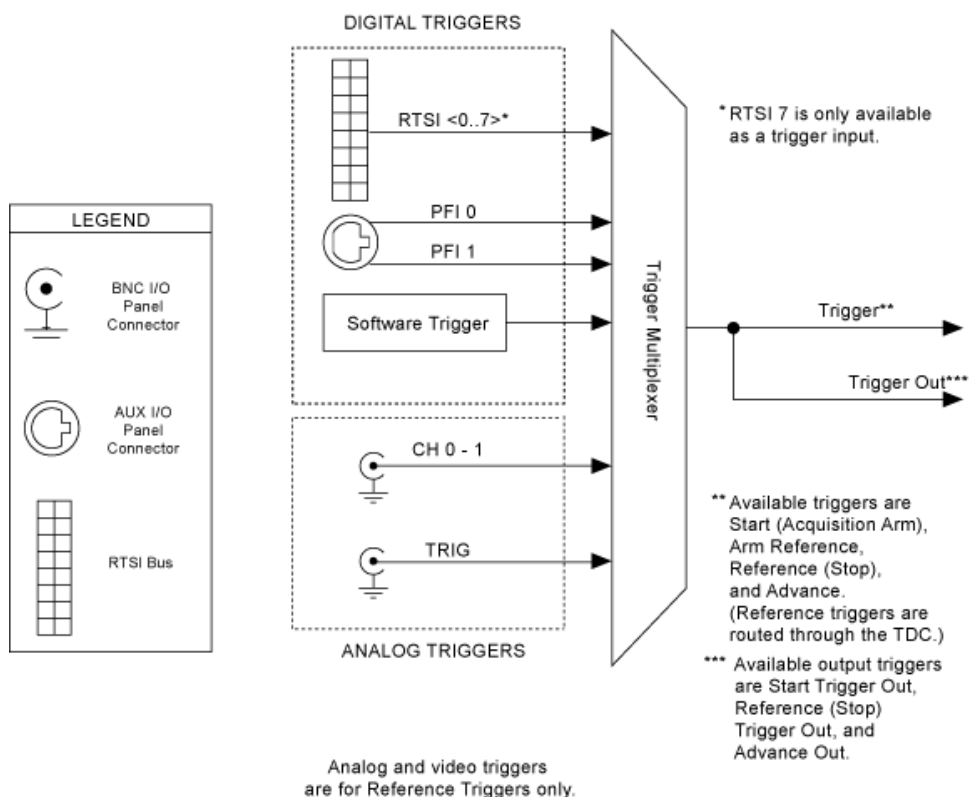
NI PXI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5114/5122/5124.



NI PCI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5114/5122/5124.



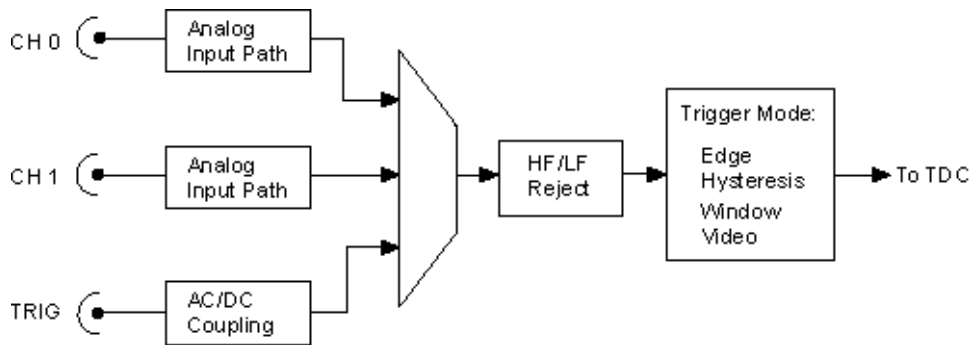
NI 5114/5122/5124/5142 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5114/5122/5124/5142 provides flexible, high-precision, low-jitter triggering features.

The NI 5114/5122/5124/5142 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5114/5122/5124/5142.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including

overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI 5114/5152/5153/5154 Trigger Delay

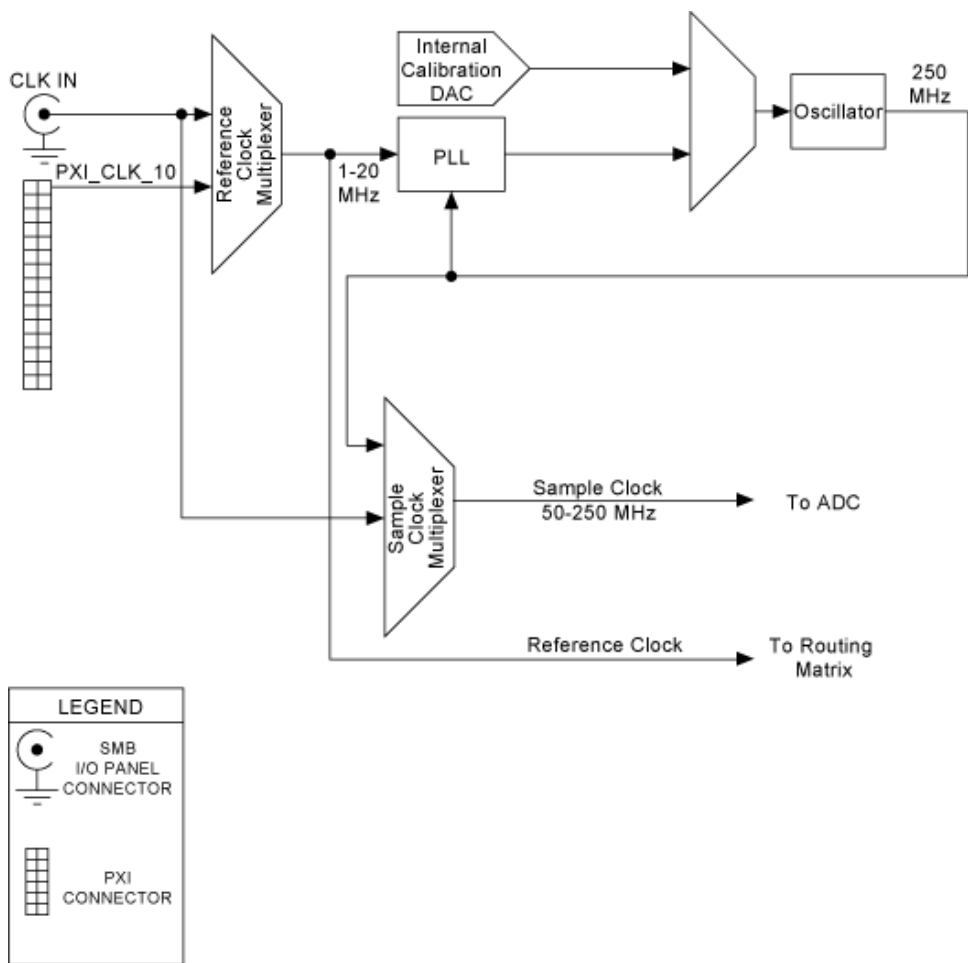
[Trigger delay](#), which is specified in seconds, is achieved on the NI 5114/5152/5153/5154 by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

Max trigger delay in seconds = $[(2^{35} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$

Note The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock.

NI PXIe-5114 Clocking

The clock circuitry on the NI PXIe-5114 offers versatile clocking options with its ability to use either the internal 250 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PXIe-5114 to phase lock the internal 250 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXIe-5114.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PXIe-5114 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PXIe-5114 has an onboard voltage controlled crystal oscillator (VCXO) running at 250 MHz. When using the onboard 250 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 250 MHz frequency of the VCXO. In PLL mode, the NI PXIe-5114 [phase locks](#) its 250 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the NI PXIe-5114 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 250 MHz clock. In these cases the NI PXIe-5114 can accept an external sample clock. External clocking also provides a method to synchronize the NI PXIe-5114 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PXIe-5114 from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the **NI PXI/PXIe/PCI-5114 Specifications** document that shipped with the device for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the **NI PXI/PXIe/PCI-5114 Specifications** for information on how this affects trigger holdoff and trigger resolution.

[Reference Clock](#)

The [reference clock](#) is used in the NI PXIe-5114 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The NI PXIe-5114 can accept a reference clock from its front panel (CLK IN) as well as from PXI_CLK10. This

reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PXIe-5114 reference clock is `None`, or no reference clock is used.



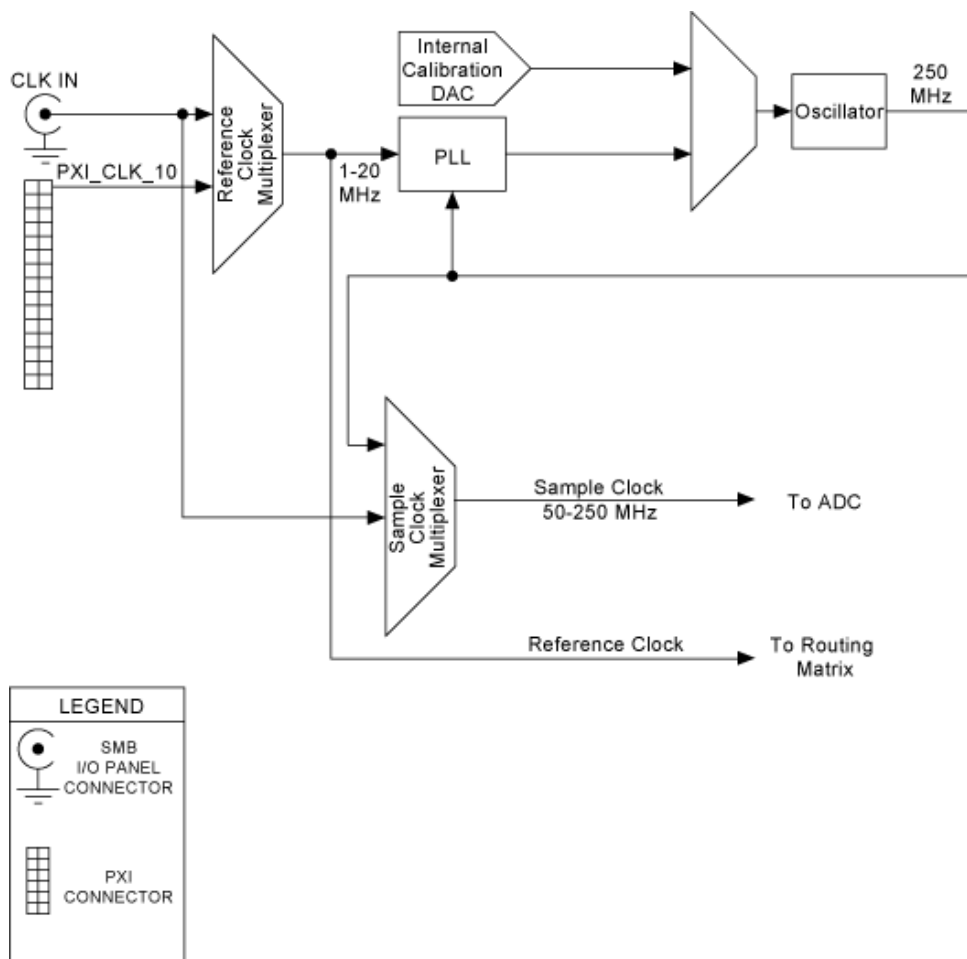
Note Locking to a reference clock is not valid when using an external sample clock.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXIe-5114 Routing Matrix](#).

NI PXI-5114 Clocking

The clock circuitry on the NI PXI-5114 offers versatile clocking options with its ability to use either the internal 250 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PXI-5114 to phase lock the internal 250 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXI-5114.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PXI-5114 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PXI-5114 has an onboard voltage controlled crystal oscillator (VCXO) running at 250 MHz. When using the onboard 250 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 250 MHz frequency of the VCXO. In PLL mode, the NI PXI-5114 [phase locks](#) its

250 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the NI PXI-5114 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 250 MHz clock. In these cases the NI PXI-5114 can accept an external sample clock. External clocking also provides a method to synchronize the NI PXI-5114 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PXI-5114 from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the **NI PXI/PCI-5114 Specifications** document that shipped with the device for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the **NI PXI/PCI-5114 Specifications** for information on how this affects trigger holdoff and trigger resolution.

Reference Clock

The [reference clock](#) is used in the NI PXI-5114 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The NI PXI-5114 can accept a reference clock from its front panel (CLK IN) as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PXI-5114 reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

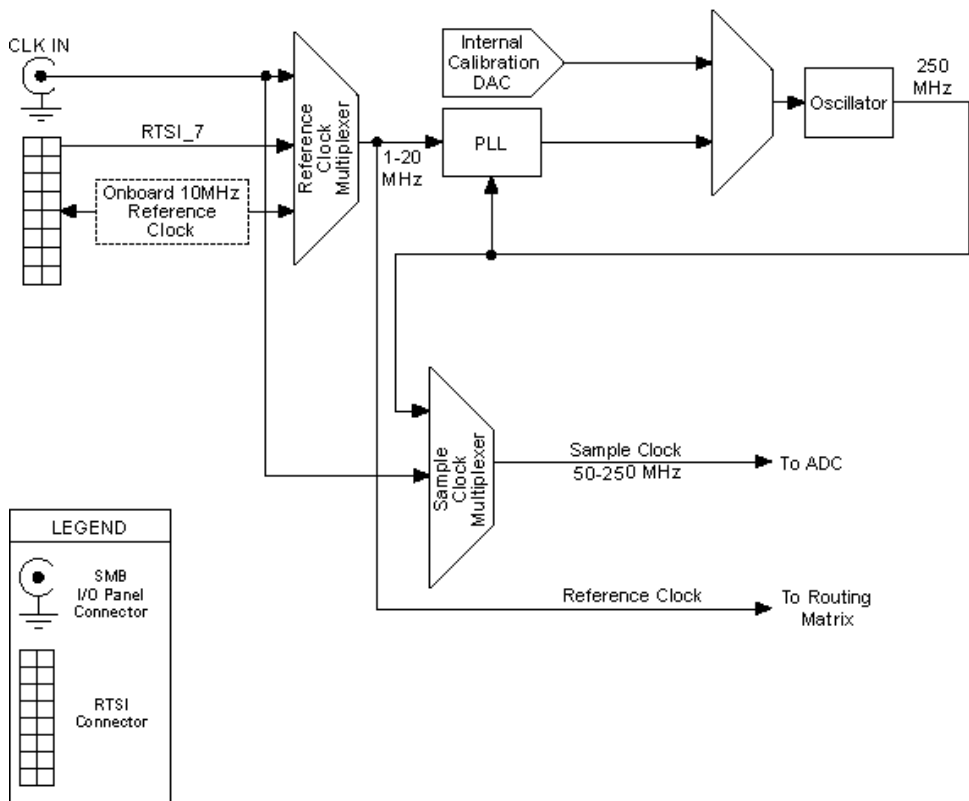
Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more

information on exporting the reference clock, refer to the [NI PXI-5114 Routing Matrix](#).

NI PCI-5114 Clocking

The clock circuitry on the NI PCI-5114 offers versatile clocking options with its ability to use either the internal 250 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PCI-5114 to phase lock the internal 250 MHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5114.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PCI-5114 can decimate its sample clock (internal or external) by an integer

divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5114 has an onboard voltage controlled crystal oscillator (VCXO) running at 250 MHz. When using the onboard 250 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 250 MHz frequency of the VCXO. In PLL mode, the NI PCI-5114 [phase locks](#) its 250 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the NI PCI-5114 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 250 MHz clock. In these cases the NI PCI-5114 can accept an external sample clock. External clocking also provides a method to synchronize the NI PCI-5114 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PCI-5114 from the front panel connector. Refer to the **NI PXI/PCI-5114 Specifications** document that shipped with the device for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the **NI PXI/PCI-5114 Specifications** for information on how this affects trigger holdoff and trigger resolution.

[Reference Clock](#)

The [reference clock](#) is used in the NI PCI-5114 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The NI PCI-5114 can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) or the internal 10 MHz reference clock. This reference clock can be any frequency from 5 MHz to 20 MHz if it is provided to CLK IN. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The

default setting for the NI PCI-5114 reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5114 Routing Matrix](#).

NI 5114 Onboard Memory

The NI 5114 allocates at least 256 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is 256 8-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes, but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the size of the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 256 samples. The available memory options are 8 MS, 64 MS, and 256 MS per channel.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5114 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition.

However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

NI 5114 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV}$ for temperature range 18-28 °C

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple instrument integrated together, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to achieve the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC), error is specified as

$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C}$ (a typical number is 10% of accuracy/ $^\circ\text{C}$)

the additional error is

$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV})$ or 120 mV

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 $^\circ\text{C}$ ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5114, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges, both input impedance paths (50 Ω and 1 M Ω), and all filter paths (enabled/disabled).
- Calibrates trigger level offset and gain.
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Compensates 1 M Ω input frequency flatness.
- Takes approximately 1 minute to complete.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 $^\circ\text{C}$ from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5114 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

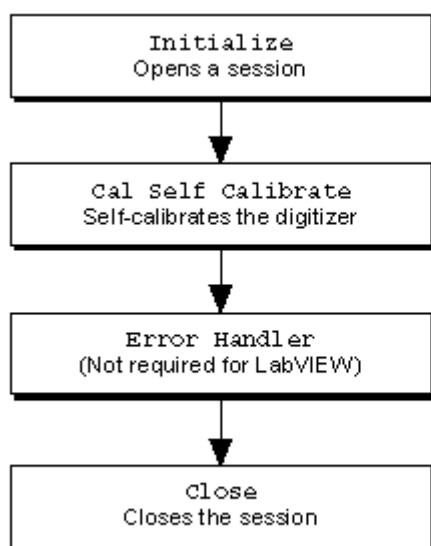
Input Connections During Self-Calibration

The NI 5114 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
-------------	--------	------	-------

External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger level Trigger timing AC flatness Input capacitance	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy Optimizes performance with external 10:1 probes
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

NI 5122

NI 5122 high-speed digitizers feature large dynamic range, wide analog bandwidth, and deep onboard acquisition memory for both time and frequency domain measurements.

Features

The NI 5122 has the following features:

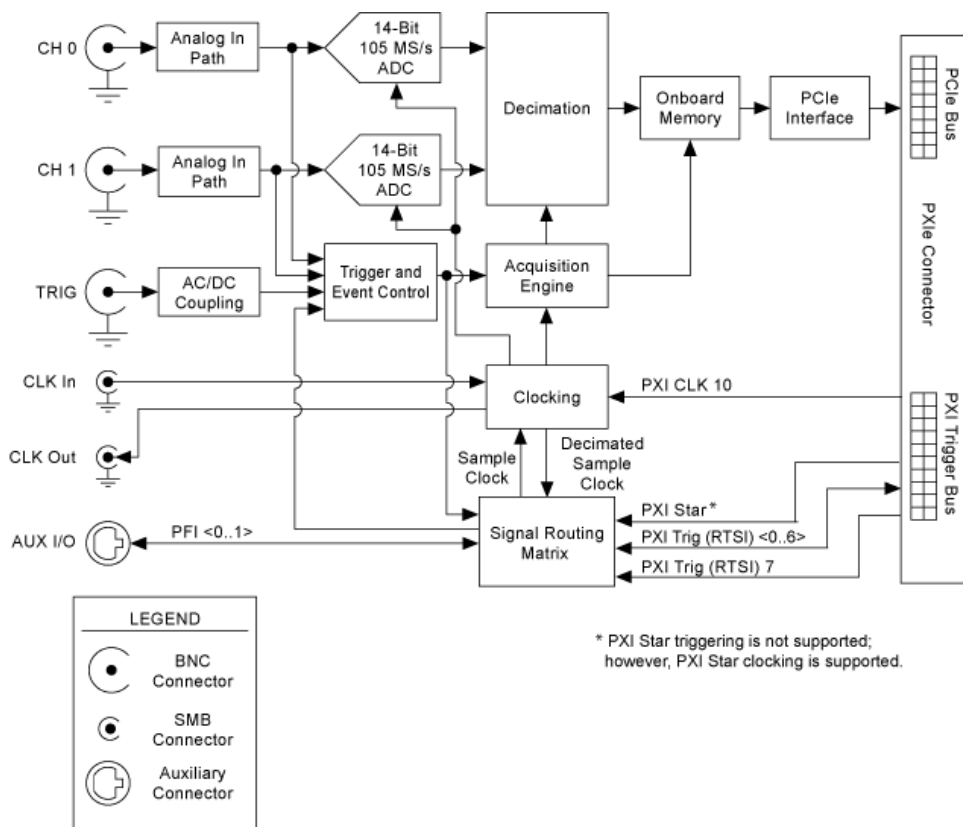
- 2 channels, simultaneously sampled
- 14-bit vertical resolution
- 100 MS/s real-time sampling rate
- 2.0 GS/s random interleaved sampling (RIS) rate
- 100 MHz bandwidth
- PXI, PXI Express, or PCI versions
- 8, 32, 256, or 512 MB (4, 16, 128, or 256 M 14-bit samples) of memory per channel, depending on memory option purchased
- [NI-TClk synchronization](#)

Related Topics

Features Supported by Device

NI PXIe-5122 Block Diagram

The following figure shows a detailed block diagram of the NI PXIe-5122.

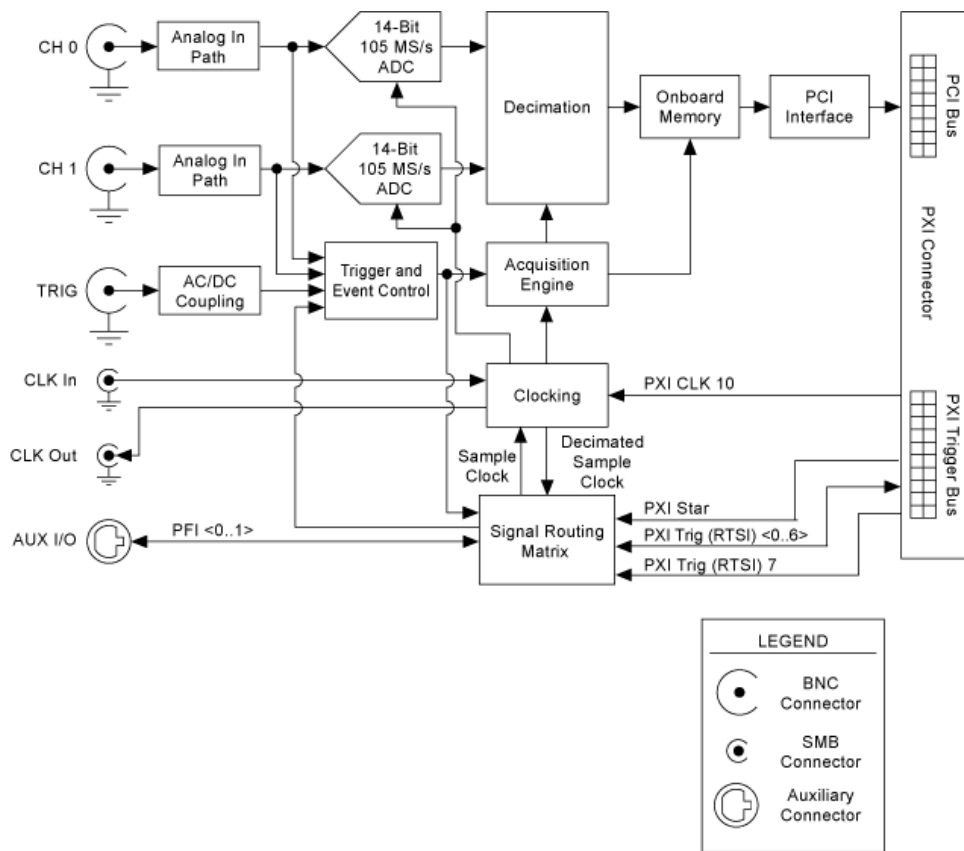


Related Web Topics

PXI Express Specification Tutorial

NI PXI-5122 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5122.

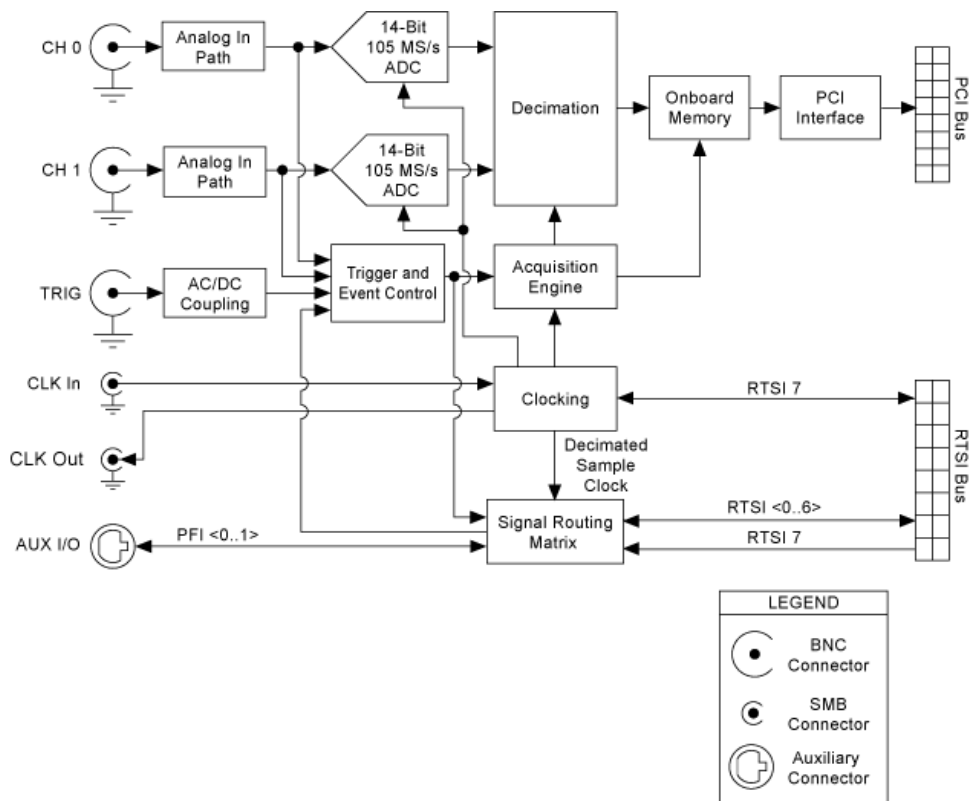


Related Web Topic

[PXI Specification Tutorial](#)

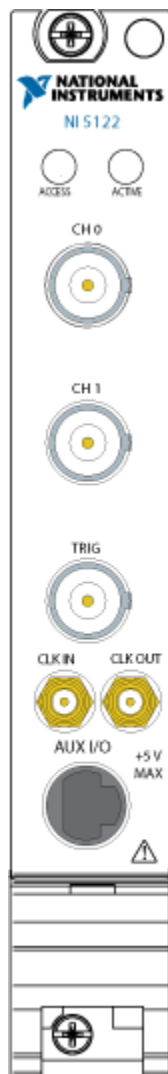
[NI PCI-5122 Block Diagram](#)

The following figure shows a detailed block diagram of the NI PCI-5122.



NI PXI/PXIe-5122 Front Panel

The following figure shows the front panel of the NI PXI/PXIe-5122. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI PXI/PXIe-5122 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
Red	The module has detected an overheating error, and you must cycle the power to your system.

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Reference (Stop) Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—the module is unable to detect the external sample clock. ▪ Overheating error—the module has detected an overheating error. Refer to Thermal Shutdown for more information. <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p>

Connectors

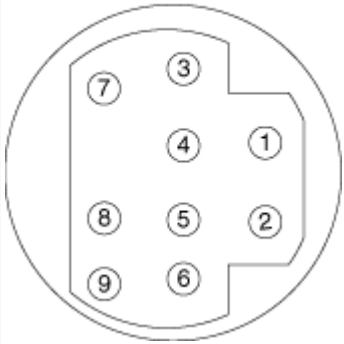
The NI 5122 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions
---------------------	--



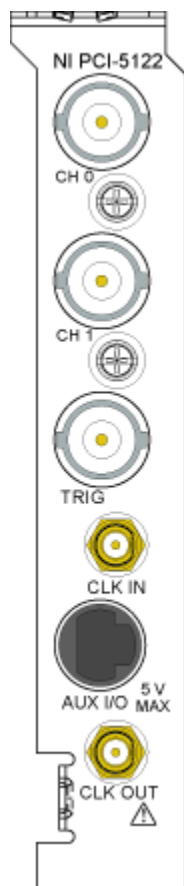
Pin Number	Description
1	5 V (Fused)
2	GND
3	Reserved
4	Reserved
5	Reserved
6	PFI 1
7	Reserved
8	Reserved
9	PFI 0



Note Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PCI-5122 Front Panel

The following figure shows the front panel of the NI PCI-5122. Descriptions of the connectors are shown below.



Connectors

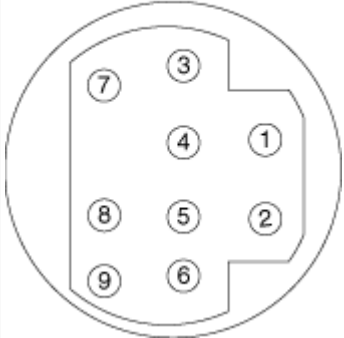
The NI PCI-5122 has the following six connectors on the front panel.


Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock

AUX I/O	9-pin mini-circular DIN connect or	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)
---------	------------------------------------	--

AUX I/O Connector Pin Assignments

PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that

shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this

value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

Features Supported by SMC-Based Devices and USB Devices

NI PXIe-5122 Cold Temperature Considerations

The NI PXIe-5122 may have special considerations for maximum data transfer rates when operating in ambient temperatures between 0 and 10 °C.

PXI Express uses PCI Express signaling and protocols for communication. Refer to [PXI Express Topology](#) for more information about the components of a PCI Express system.

NI PXIe-5122 Considerations

The NI PXIe-5122 supports x4, x2 or x1 links. The NI PXIe-5122 always uses 2.5 Gb/s signaling rates. When the digitizer is installed in a PXI Express slot that uses a x4 (or x8 or wider) PCIe switch, the NI PXIe-5122 and the switch train to form a x4 link. Currently all NI PXI Express chassis use x4 switches for all PXI Express slots. If the NI PXIe-5122 is installed in a system that uses a x2 or x1 switch, the NI PXIe-5122 trains to form a x2 or x1 link.

When the NI PXIe-5122 trains with a x4 lane width, the maximum theoretical throughput of the link is 1,000 MB/s. Note that the NI PXI-5122 generates data at 400 MB/s (two channels at 100 MS/s at two bytes per sample).

If the NI PXIe-5122 trains as x2 or x1, the NI PXIe-5122 still functions properly; and all measurement specifications are unchanged. However, the maximum theoretical throughput of the link is decreased to 500 MB/s (for x2) or 250 MB/s (for x1). Again the actual transfer rate to host memory will be slower and depends on many other aspects of the system. The NI PXIe-5122 can still acquire data and store to onboard memory at the full specified rate of 100 MS/s on two channels.

Booting NI PXIe-5122 at cold temperatures


In rare cases, when a PXI system is turned on and the ambient temperature is below 10 °C, the NI PXIe-5122 may detect a sufficient number of bit errors during training,


that the link reverts to a x2 or x1 width. If this occurs, you can still acquire data to onboard memory at the full specified acquisition rate of 100 MS/s on two channels. To retrain as a x4 width link in this case, complete the following steps:

1. Keep the system powered on for 15 minutes (the specified warm-up time for the analog specifications to be valid for the device).
2. Reboot.

Detecting Lane Widths

NI provides two utilities (`GetPXIEExpressLinkWidth.exe` and `GetPXIEExpressLinkWidth.vi`) that can detect lane widths and display the lane width currently in use by the device. Refer to "What are the recommendations for using my PXIe-5122 below 10 °C?" to download one of these utilities.

 **Note** Make sure NI-SCOPE version 3.3.1 or later is installed in the system before running these utilities.

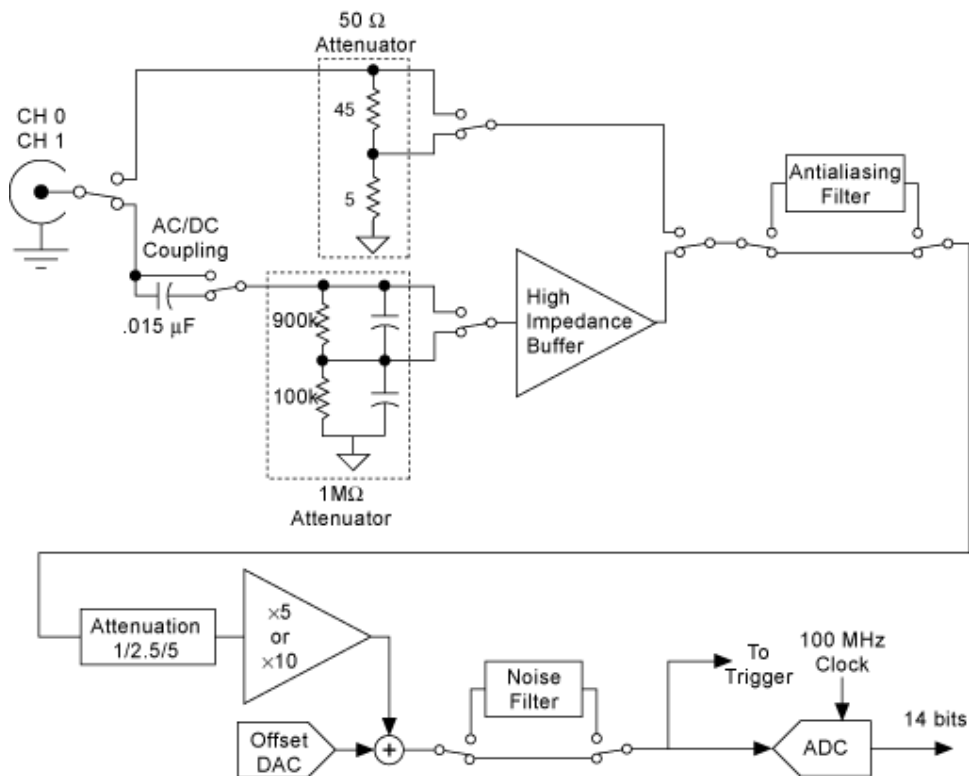
 **Note** This utility should only be run on a system that supports link widths of 4. The NI PXIe-8103 controller, for example, is a x1 controller and the NI 5122 will always be trained as a x1 with this controller. The same is true with x1 MXI Express.

Related Topic

[PXI Express Topology](#)

NI 5122/5142 Input Signal Conditioning

The NI 5122/5142 provides two independent digitizer input channel signal conditioning paths. Each path provides you with a choice of 50 Ω input impedance or 1 M Ω input impedance, as shown in the following diagram.



The 50 Ω path is optimized for flatness, distortion, dynamic range, and noise because the active circuitry in the signal path is minimized. In some applications, this optimization is the primary consideration. High-performance communications measurements can take advantage of the large SFDR and 14-bit dynamic range of the NI 5122/5142.

The 1 MΩ path provides the traditional oscilloscope input impedance and loading characteristics for numerous general-purpose measurement applications.

Note The ground on the device inputs is connected to the chassis ground.

NI 5122/5124/5142 Input Ranges

As shown in the following table, the 1 MΩ path allows up to 20 V_{pk-pk} whereas the 50 Ω path is limited to 10 V_{pk-pk}.

50Ω Input Path	1MΩ Input Path
0.2 V _{pk-pk}	0.2 V _{pk-pk}
0.4 V _{pk-pk}	0.4 V _{pk-pk}

1 V _{pk-pk}	1 V _{pk-pk}
2 V _{pk-pk}	2 V _{pk-pk}
4 V _{pk-pk}	4 V _{pk-pk}
10 V _{pk-pk}	10 V _{pk-pk}
—	20 V _{pk-pk}



Note Except for the highest ranges (10 V for 50 Ω, 20 V for 1 MΩ), you can set the [vertical offset](#) on the NI 5122/5124/5142 to effectively create a unipolar range.

NI 5122/5124/5142 Input Impedance

You can set the NI 5122/5124/5142/5152 analog input impedance to either 50 Ω or 1 MΩ. The 50 Ω path provides slightly better noise and distortion performance because the high impedance buffer for the 1 MΩ path is bypassed. The 1 MΩ path is required in applications that require minimal loading or that require using a standard 10:1 oscilloscope probe.

Protection

The 50 Ω inputs of the NI 5122/5124/5142 are protected by a thermal disconnect circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω.

NI 5114/5122/5124/5142 AC/DC/GND Coupling

You can select AC, DC, or GND [input coupling](#) for the 1 MΩ input path, and DC or GND input coupling for the 50 Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit, or if you are using the 50 Ω path.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5114/5122/5124/5142 Vertical Offset

On NI 5114/5122/5124/5142 devices, you can set a hardware-applied vertical offset up to one-half of the given range, which effectively creates a unipolar range. For example, if you set the vertical offset to -1 V for the 2 V range, you can measure input voltages between 0 and -2 V.



Note Vertical offset is not available on the 20 V range for 1 M Ω , or the 10 V range for 50 Ω .

NI 5122/5142 Filters

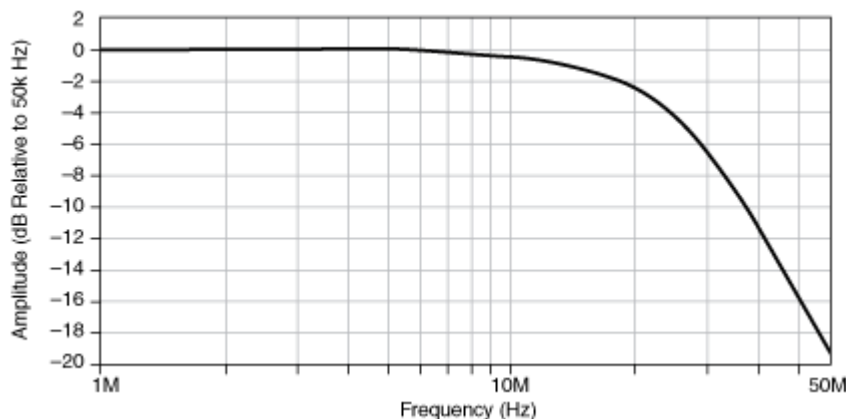
The NI 5122/5142 offers two filters that limit the bandwidth of the signal path—a 20 MHz noise filter and a 35 MHz antialias filter.



Note To change the filter type, call [niScope Configure Chan Characteristics VI](#) or the [niScope_ConfigureChanCharacteristics](#) function and set the Max Input Frequency parameter to the appropriate value.

Noise Filter

A 20 MHz input filter is available through both the 1 M Ω and 50 Ω signal paths. This filter is intended to reduce noise when the signal content is 20 MHz or less. A typical frequency response of the noise filter is shown in the following figure.

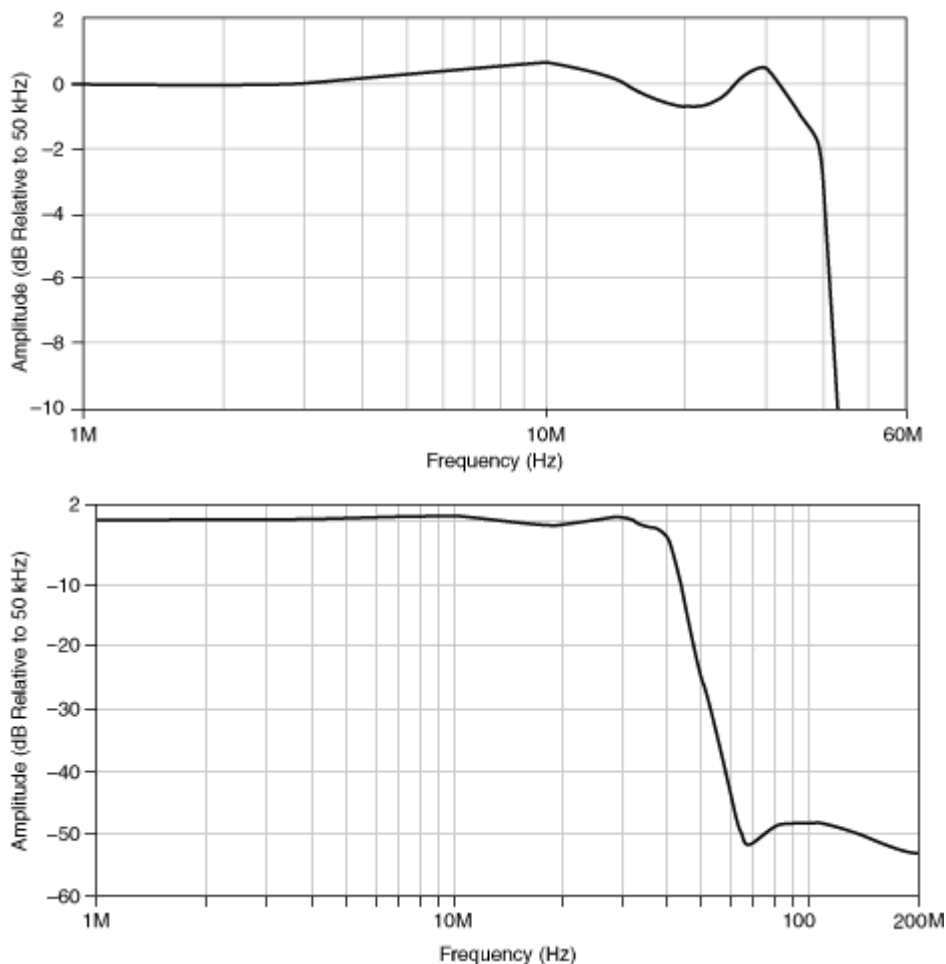


The noise filter provides over 20 dB of attenuation at 100 MHz, which is useful for rejecting out of band or background noise in lower frequency applications. An example application would be characterization of video waveforms in the presence of IF noise. Because most of the signal content is below the 8 MHz range, the 20 MHz noise filter is ideal. Many standard noise tests specify a 20 MHz noise bandwidth.

Antialias Filter

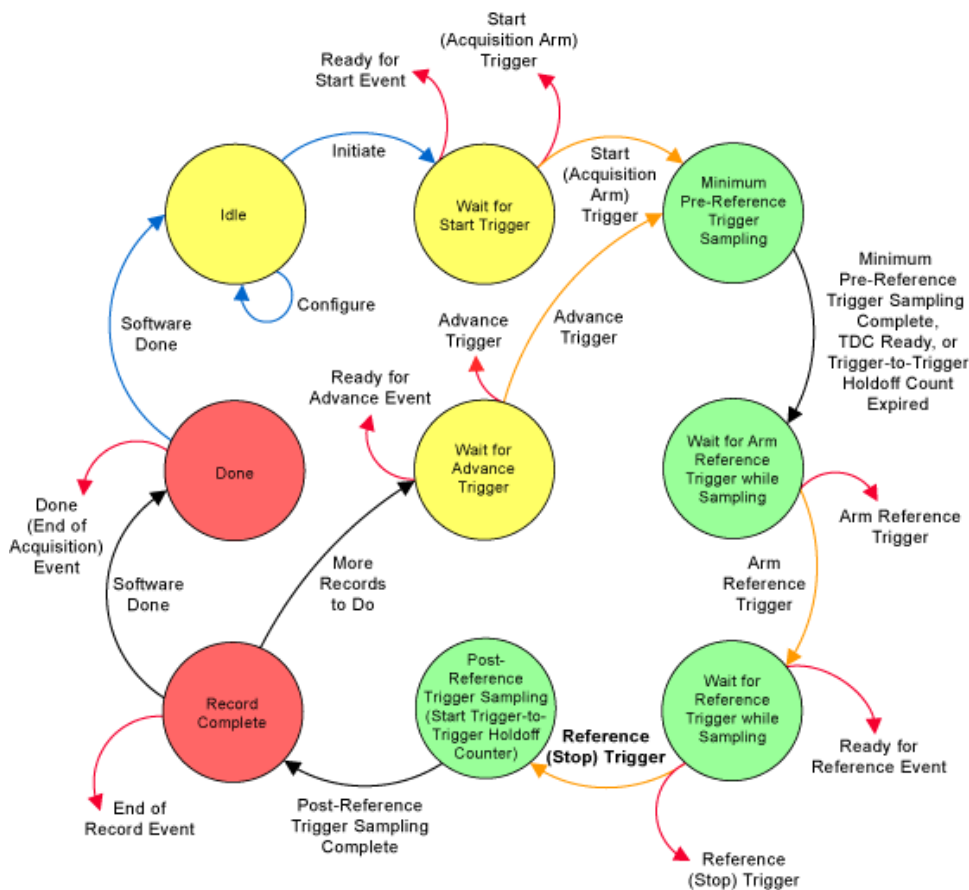
The NI 5122/5142 provides another bandwidth-limiting filter called an antialias filter. Although this filter can be used to limit out of band noise in many applications, its primary purpose is to minimize [aliasing](#) effects caused by signals greater than [Nyquist](#) applied to the 100 MS/s ADC. Refer to the [hardware specifications document](#) for more information on this filter.


A typical response with the antialias filter is shown in the following two figures.



SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



 **Note** The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

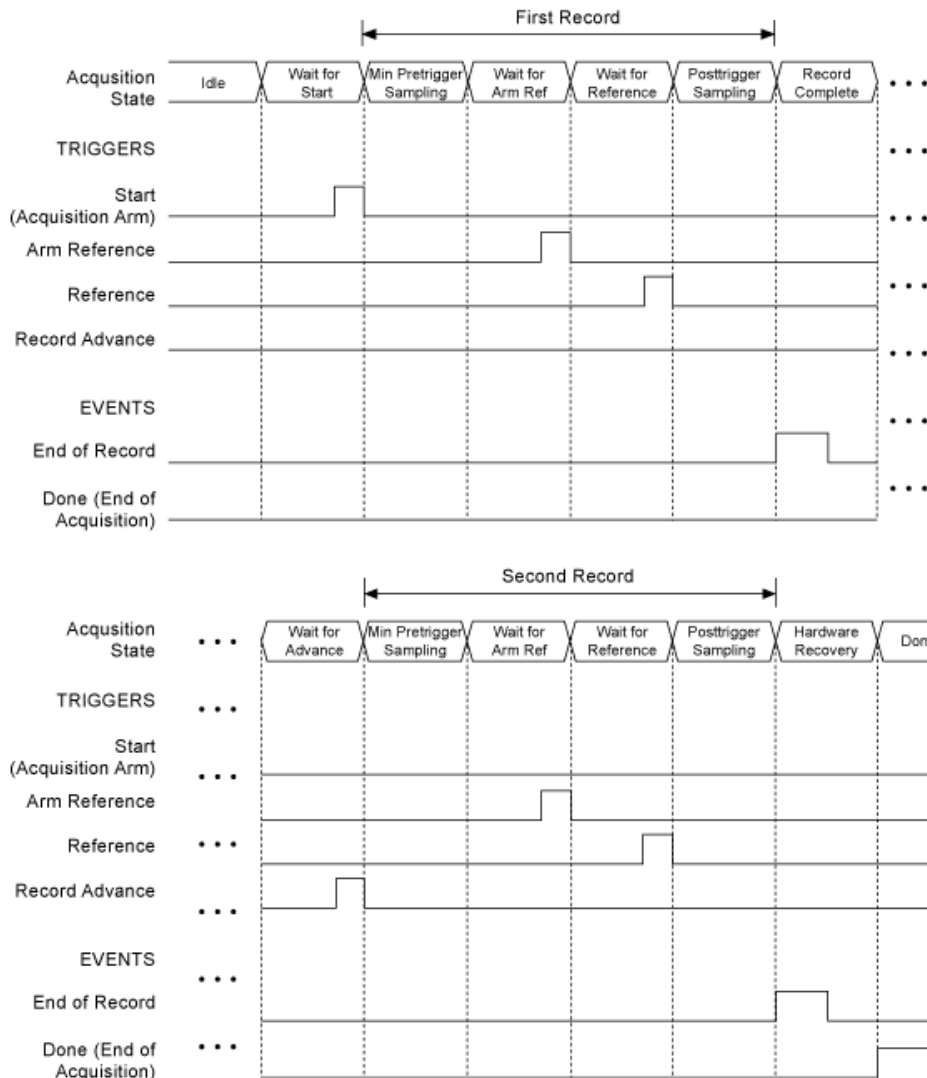
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5122/5124/5142 and NI PXIe-5122 Routing Matrix

The following table shows the signals available for export from the NI PXI-5122/5124/5142 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)
Exported Clocks			
Reference Clock (External)	√	√	√
Sample Clock (Full Rate)	—	√	—
Sample Clock (Decimated Rate)	√	√	√
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

NI PCI-5122/5124/5142 Routing Matrix

The following table shows the signals available for export from the NI PCI-5122/5124/5142, and the lines to which they can be routed.

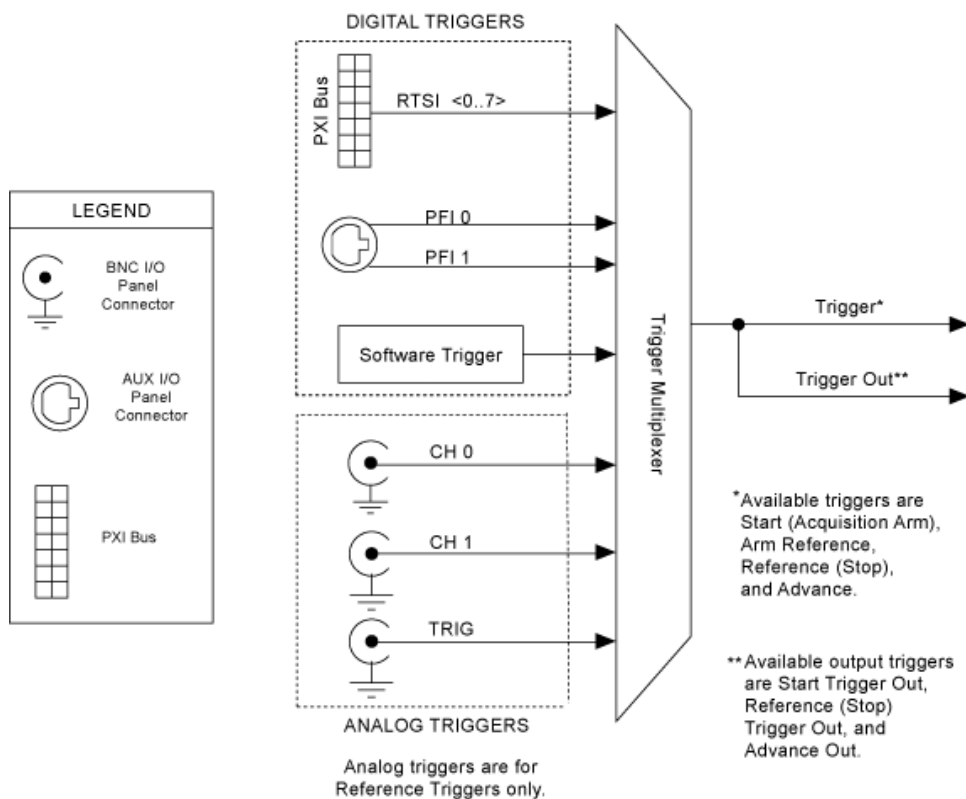
Source	Destination			
	RTSI <0..6> (RTSI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)	RTSI 7
Exported Clocks				
Reference Clock (External)	—	√	√	—
Reference Clock (Internal)	—	√	√	√
Sample Clock (Full Rate)	—	√	—	—
Sample Clock (Decimated Rate)	√	√	√	—
Triggers				
Acquisition Arm (Start) Trigger	√	—	√	—
Reference (Stop) Trigger	√	—	√	—
Events				
End of Record Event	√	—	√	—
End of Acquisition Event	√	—	√	—
Ready for Start Event	√	—	√	—
Ready for Reference Event	√	—	√	—
Ready for Advance Event	√	—	√	—

NI 5114/5122/5124/5142 Analog Trigger Types

The NI 5114/5122/5124/5142 supports the following analog trigger types: [edge](#), [hysteresis](#), [window](#), and [video](#) triggers.

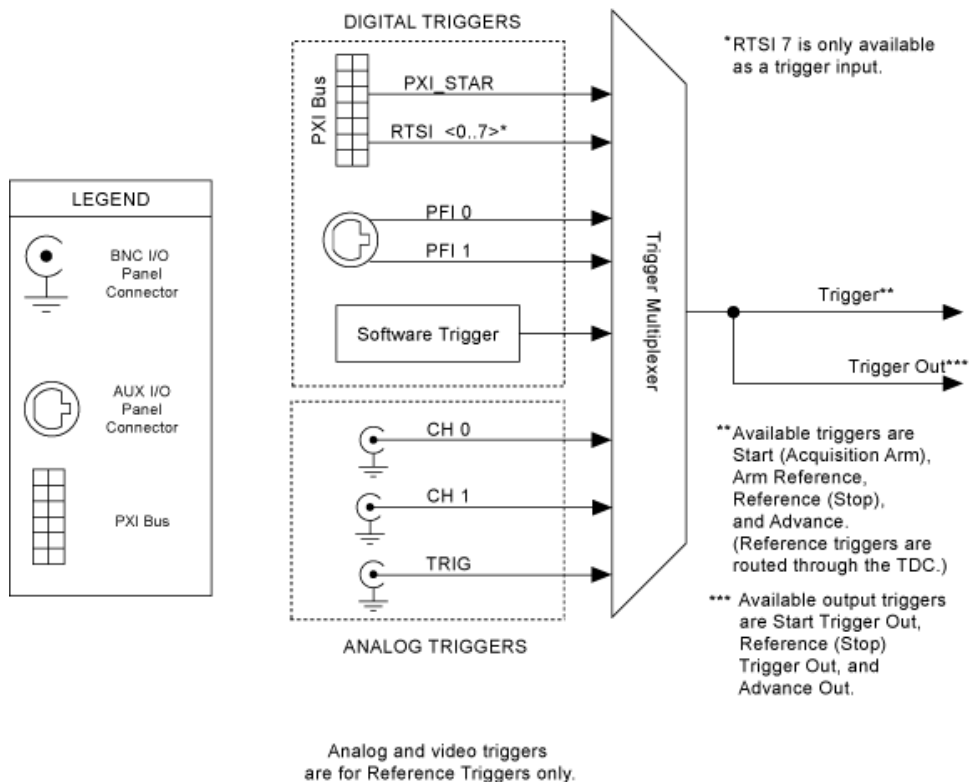
NI PXIe-5122 Trigger Sources

The following figure shows the trigger sources for the NI PXIe-5122.



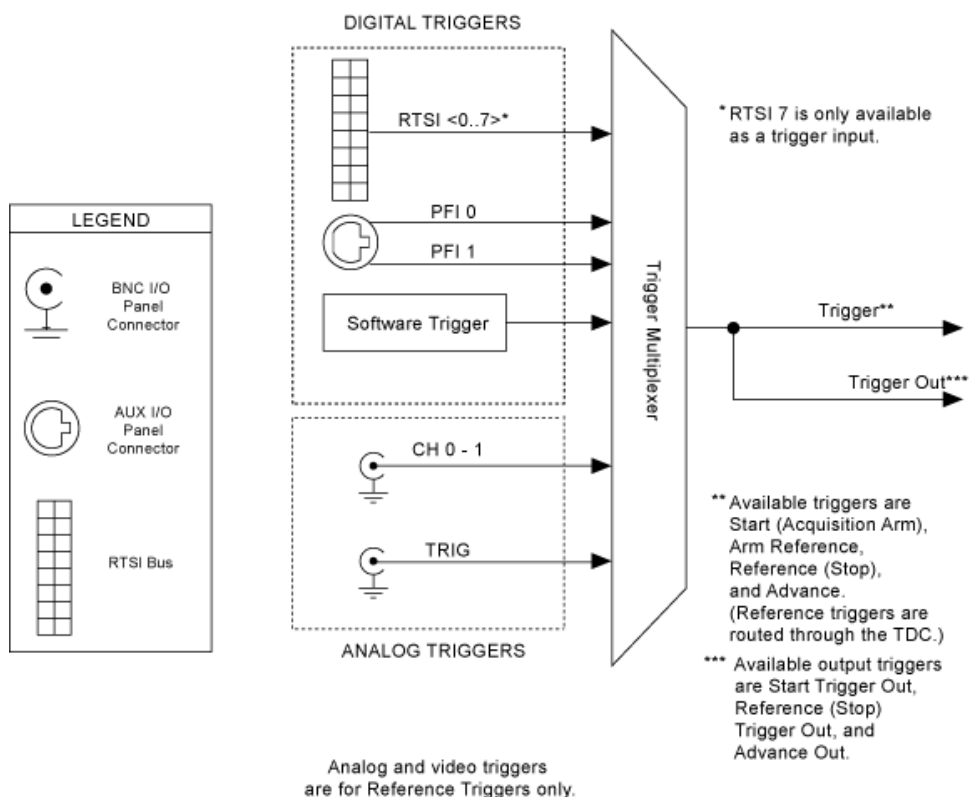
NI PXI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5114/5122/5124.



NI PCI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5114/5122/5124.



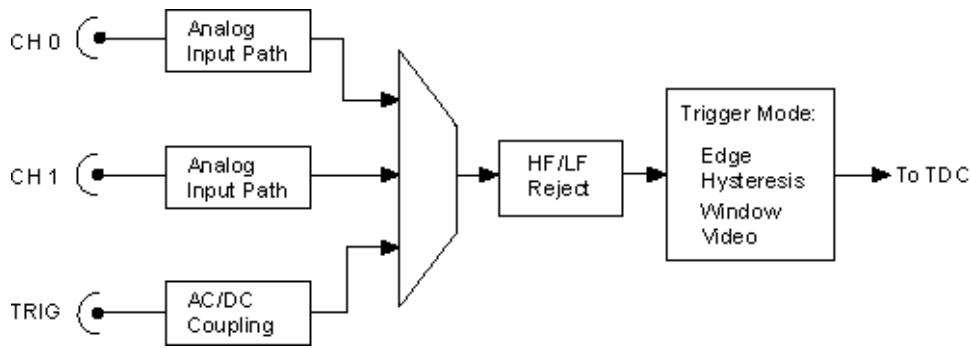
NI 5114/5122/5124/5142 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5114/5122/5124/5142 provides flexible, high-precision, low-jitter triggering features.

The NI 5114/5122/5124/5142 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5114/5122/5124/5142.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including

overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI 5122/5124/5142/5622/5922 Trigger Delay

[Trigger delay](#), which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

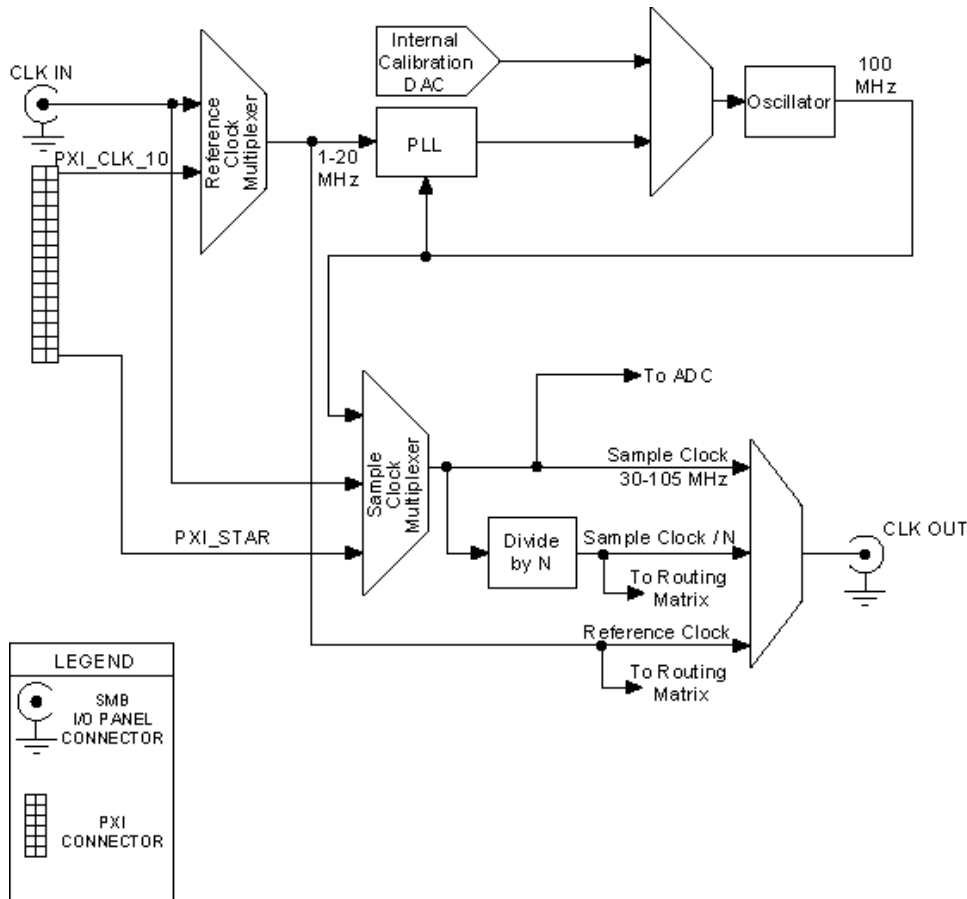
Max trigger delay in seconds = $[(2^{34} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock. The maximum trigger delay also changes based on the binary sample width. When binary sample width is set to 8, the maximum trigger delay is doubled. When binary sample width is set to 32, the maximum trigger delay is halved.

NI PXI-5122/5142 and NI PXIe-5122 Clocking

The clock circuitry on the NI PXI-5122/5142 and the NI PXIe-5122 offer versatile clocking options with the ability to use either the internal 100 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on these digitizers to phase lock the internal 100 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of these digitizers.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The digitizers can decimate their sample clocks (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

These digitizers have an onboard voltage controlled crystal oscillator (VCXO) running at 100 MHz. When using the onboard 100 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 100 MHz frequency of the VCXO. In PLL mode, the device phase locks its 100 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 100 MHz clock. In these cases these digitizers can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The [reference clock](#) is used in the digitizer's phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizers can accept a reference clock from the front panel (CLK IN) as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 100 MS/s when using the internal sample clock, the digitizer decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PXI-5122/5142 can only store one out of every two samples received from the ADCs to onboard memory. They can sample at rates of $100/n$ MS/s, where n is an integer value between 1 and 2^{16} . Because the digitizer does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When sampling at the

maximum rate, a free-running version of the actual sample clock is exported to the front panel of the digitizer. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.



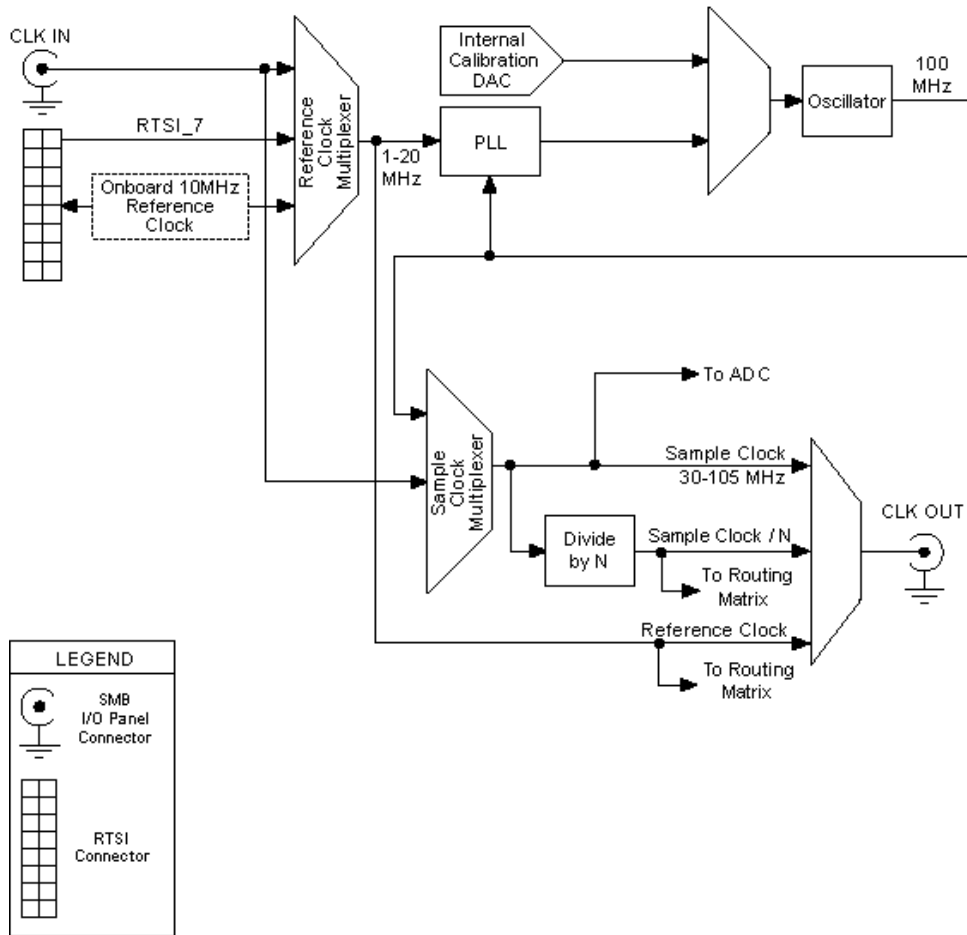
Note The NI PXI-5142 cannot export its sample clock while operating in [DDC](#) mode. If you want to export the sample clock, set the [DDC Enabled](#) property to False (default).

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXI-5122/5124/5142 and NI PXIe-5122 Routing Matrix](#).

NI PCI-5122/5142 Clocking

The clock circuitry on the NI PCI-5122/5142 offers versatile clocking options with its ability to use either the internal 100 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PCI-5122/5142 to phase lock the internal 100 MHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5122/5142.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PCI-5122/5142 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5122/5142 has an onboard voltage controlled crystal oscillator (VCXO) running at 100 MHz. When using the onboard 100 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 100 MHz frequency of the VCXO. In PLL mode, the NI PCI-5122/5142 phase locks its 100 MHz sample clock to the supplied reference clock. The PLL mode is

useful when synchronizing the NI PCI-5122/5142 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 100 MHz clock. In these cases the NI PCI-5122/5142 can accept an external sample clock. External clocking also provides a method to synchronize the NI PCI-5122/5142 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PCI-5122/5142 from the front panel connector. Refer to the specifications document for your device for external sample clock requirements.



Note When an external sample clock is used, the [TDC](#) is disabled. Refer to the device specifications for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The reference clock is used in the NI PCI-5122/5142 PLL circuit to synchronize the sample clock to the reference clock. The NI PCI-5122/5142 can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 5 MHz to 20 MHz if it is provided to CLK IN or to RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PCI-5122/5142 reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 100 MS/s when using the internal sample clock, the NI PCI-5122/5142 decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PCI-5122/5142 only stores one out of every two samples it receives from the ADCs to onboard memory. The NI PCI-5122/5142 can sample at rates of $100/n$ MS/s, where n is an integer value between 1 and (2^{16}) . Because the

NI PCI-5122/5142 does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When sampling at the maximum rate, a free-running version of the actual sample clock is exported to the front panel of the NI PCI-5122/5142. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.

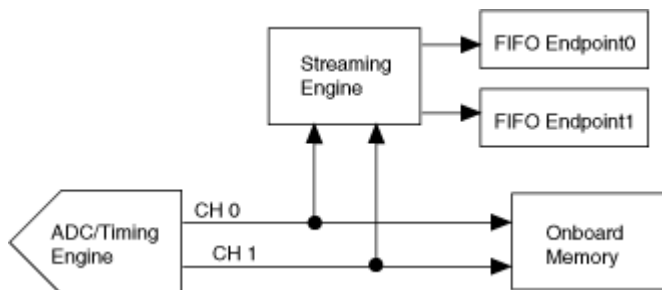
Exporting Reference Clock

If you are using the internal reference clock you can export that same reference clock to other NI PCI-5122/5142 devices for synchronization purposes. For more information, refer to the section on synchronization. If you are using a reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5122/5124/5142 Routing Matrix](#).

NI PXIe-5122 Peer-to-Peer Streaming

The NI PXIe-5122 supports peer-to-peer streaming using the NI-P2P API. For more information about using NI-P2P to stream data between devices, refer to the **Peer-to-Peer Streaming Help**. The [NI-SCOPE Function Reference Help](#) and the [NI-SCOPE LabVIEW Reference Help](#) provide more information on configuring the digitizer's endpoint resources to send data to another peer.

The digitizer's two writer endpoints are located on parallel data paths to the onboard memory. Acquired data can be written to these endpoints as well as fetched to the host through onboard memory. By default, data is not written to onboard memory while any peer-to-peer endpoint is enabled. Set the [Onboard Memory Enabled](#) property or the [NISCOPE_ATTR_P2P_ONBOARD_MEMORY_ENABLED](#) attribute to store data to onboard memory while a peer-to-peer endpoint is enabled.



NI PXIe-5122 Configuring an Endpoint

To set properties on an endpoint using LabVIEW, you must select that endpoint using the [Active Channel](#) property and the endpoint name FIFOEndpoint0 or FIFOEndpoint1.

Considerations When Configuring an Endpoint

Keep the following points in mind when configuring an endpoint:

- Set the [P2P Enabled](#) property or the [NISCOPE_ATTR_P2P_ENABLED](#) attribute to TRUE to activate the P2P endpoint and to use any of the P2P features.
- Use the [Channels to Stream](#) property or the [NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM](#) to determine the specific data that is sent to an endpoint.
- Use the [Binary Sample Width](#) property or the [NISCOPE_ATTR_BINARY_SAMPLE_WIDTH](#) attribute to set the sample width.
- Call niScope Abort, niScope Close, or niScope Reset on the digitizer session to reset an endpoint.

FIFOEndpoint0 Channels to Stream	FIFOEndpoint1 Channels to Stream	Description
0	1	Data from Channel 0 is sent to FIFOEndpoint0, while data from Channel 1 is sent to FIFOEndpoint1.
0	0	A copy of the data from Channel 0 is sent to both endpoints.

1	1	A copy of the data from Channel 1 is sent to both endpoints.
0, 1	0, 1	Data from both channels is interleaved and sent to both endpoints.

Configuring a Peer-to-Peer Stream

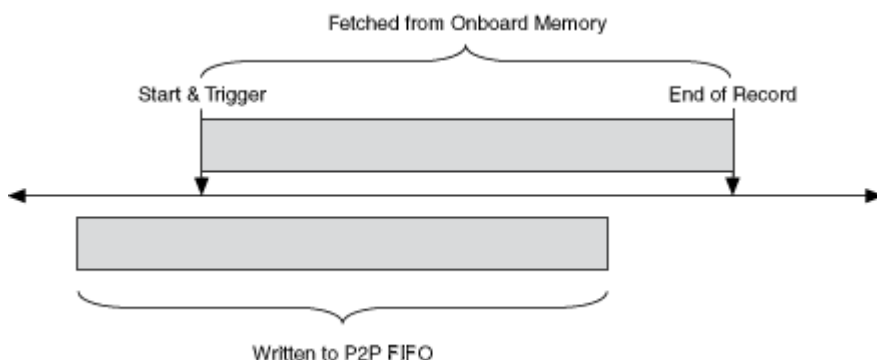
To configure a peer-to-peer stream using the NI-P2P API, a writer and reader handle for each endpoint are required. Use the [niScope Get Stream Endpoint Handle VI](#) to get a writer endpoint handle. Refer to the **Peer-to-Peer Streaming Help** for more information on creating and monitoring peer-to-peer streams.

Configuring a Trigger with a P2P Stream

Depending on the type of trigger configured for the acquisition, data is not written to the P2P endpoints in the same way as it is written to onboard memory.

Immediate Trigger

Because of differences in the data path leading to the P2P endpoints and the onboard memory, the first sample written to the P2P endpoints is from an earlier time than the first sample written to onboard memory.

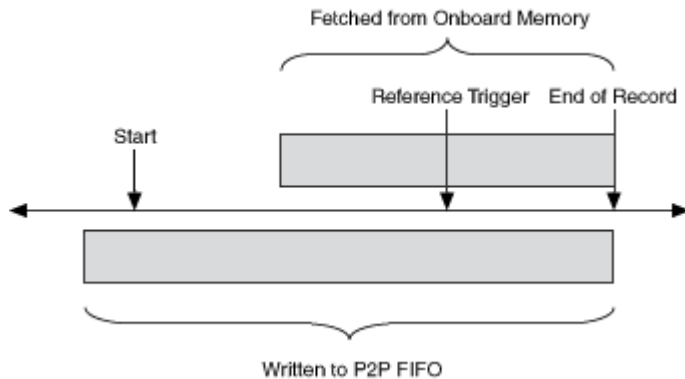


Note The record sent to the onboard memory is the same length as the record sent to the P2P FIFO.

All data written to the P2P endpoints is valid, but the samples are not aligned with the same record stored in onboard memory.

Configured Reference Trigger

When the trigger type is anything other than immediate, the digitizer sends all data between the Start trigger and the End of Record event to the endpoint.



Scaling and Normalization of Binary Data

In a normal acquisition, you cannot assume that the binary data maps perfectly to the vertical range. For example, if you fetch 16-bit binary data, you might expect that the positive full scale binary value corresponds to the maximum positive voltage of the vertical range, and that the negative full scale binary value corresponds to the maximum negative voltage. This is not the case, however, for a number of reasons.

First, the full scale of the ADC is not mapped to the vertical range. A few codes on the positive and negative ends are left as headroom, in case the input signal slightly exceeds the specified vertical range. Second, the binary values do not account for the required corrections to gain and offset based on calibration data. If you fetch binary data, you can convert to voltage values using the gain and offset values from the `wfmInfo` struct or use the [niScope Get Scaling Coefficients VI](#) or the [niScope_GetScalingCoefficients](#) function to get these coefficients. These values account for the ADC normalization and calibration data.

$$\text{Voltage} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

To get binary data that accounts for both ADC normalization and calibration, use the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function. Applying gain and offset from the normalization

coefficients results in a range of binary values that maps exactly to the vertical range you configure.

$$\text{Normalized Binary Data} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

$$\text{Voltage} = \text{Normalized Binary Data} \times (\text{Range} / 2^{\text{resolution in bits}})$$

In a normal acquisition, you must know the gain and offset values to correctly convert the binary data to voltage values. After applying gain and offset values from the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function, you can assume the following:

- The maximum positive binary value maps to the maximum positive voltage of the vertical range.
- The maximum negative binary value maps to the maximum negative voltage of the vertical range.
- The vertical range is divided evenly across the possible binary values.

Peer-to-Peer Streaming VIs/Functions

LabVIEW	C/C++
Streaming Session—Use the following NI-P2P VI to configure a peer-to-peer streaming session.	
niScope Get Stream Endpoint Handle VI	niScope_GetStreamEndpointHandle
Scaling and Normalization—Data written to a peer-to-peer endpoint is not scaled or calibrated. Use the following VIs/functions to scale or normalize the binary data.	
niScope Get Wfm Coefficients (poly) VI	niScope_GetNormalizationCoefficients
	niScope_GetScalingCoefficients

Peer-to-Peer Streaming Properties/Attributes

Use the following properties and attributes to configure a peer-to-peer streaming session.

LabVIEW	C/C++
P2P Enabled	NISCOPE_ATTR_P2P_ENABLED
Channels to Stream	NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM
Endpoint Size	NISCOPE_ATTR_P2P_ENDPOINT_SIZE

<u>Samples Available In Endpoint</u>	<u>NISCOPE_ATTR_P2P_SAMPLES_AVAIL_IN_ENDPOINT</u>
<u>Most Samples Available In Endpoint</u>	<u>NISCOPE_ATTR_P2P_MOST_SAMPLES_AVAILABLE_IN_ENDPOINT</u>
<u>Samples Transferred</u>	<u>NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED</u>
<u>Samples Transferred Per Record</u>	<u>NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED_PER_RECORD</u>
<u>Endpoint Overflow</u>	<u>NISCOPE_ATTR_P2P_ENDPOINT_OVERFLOW</u>
<u>FIFO Endpoint Count</u>	<u>NISCOPE_ATTR_P2P_FIFO_ENDPOINT_COUNT</u>
<u>Stream Relative To</u>	<u>NISCOPE_ATTR_P2P_STREAM_RELATIVE_TO</u>
<p>Manual—The following property/attribute activates advanced functionality for a P2P endpoint. To use any of the P2P advanced features, this property/attribute must be set to TRUE. This attribute cannot be used for an endpoint if that endpoint is configured using NI-P2P.</p>	
<u>Manual Configuration Enabled</u>	<u>NISCOPE_ATTR_P2P_MANUAL_CONFIGURATION_ENABLED</u>
<p>Configuration—The following properties/attributes provide a way to configure a destination for endpoint data without using NI-P2P. These attributes can be used to send data to a DMA window on a third-party device, or to a memory window on the host machine.</p>	
<u>Data Transfer Permission Address</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR</u>
<u>Data Transfer Permission Address Type</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR_TYPE</u>
<u>Destination Window Address</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR</u>
<u>Destination Window Address Type</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_TYPE</u>
<u>Destination Window Size</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_SIZE</u>
<p>Notification—The following properties/attributes can be used to send a message corresponding to an event directly to a peer or to a location in host memory.</p>	
<u>Push Message On</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON</u>

<u>Message Push Address</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR</u>
<u>Message Push Address Type</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDRESS_TYPE</u>
<u>Message Push Value</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VALUE</u>

NI 5122/5142 Onboard Memory

The NI 5122/5142 allocates at least 384 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is approximately 128 14-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 128 samples. Refer to the [hardware specifications document](#) for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5122/5142 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

NI 5122/5124/5142 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$\text{TC} = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 \text{ }^\circ\text{C} \times \text{TC} = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5122/5124/5142, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges, both input impedance paths (50 Ω and 1 MΩ), and all filter paths (enabled/disabled).
- Calibrates trigger level offset and gain.
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Compensates 1 MΩ input frequency flatness, then compensates input capacitance so that it is equal regardless of selected attenuator range.
- Takes approximately 2 minutes to complete.

Using the example as a comparison, the NI 5122/5124/5142, using self-calibration and a 5 V input, will have an error at 48 °C of

$$5 \text{ V}(0.65\% \text{ of input} + 8 \text{ mV}) = 40.5 \text{ mV error}$$

This result compares favorably to the 180 mV error for a typical digitizer as described in the example without self-calibration.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5122/5124/5142 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

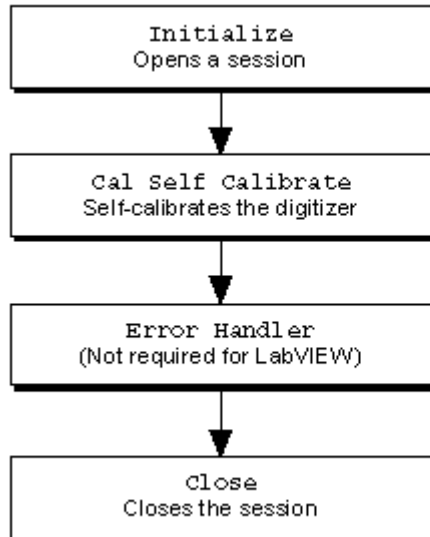
Input Connections During Self-Calibration

The NI 5122/5124/5142 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at Start»All Programs»National Instruments»NI-SCOPE»Examples.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger level Trigger timing AC flatness Input capacitance	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy Optimizes performance with external 10:1 probes
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

NI 5124

NI PXI/PCI-5124 high-speed digitizers feature large dynamic range, wide analog bandwidth, and deep onboard acquisition memory for both time and frequency domain measurements.

Features

The NI 5124 has the following features:

- 2 channels, simultaneously sampled
- 12-bit vertical resolution
- 200 MS/s real-time sampling rate
- 4.0 GS/s random interleaved sampling (RIS) rate
- 150 MHz bandwidth
- 8, 32, 256, or 512 MB of memory per channel

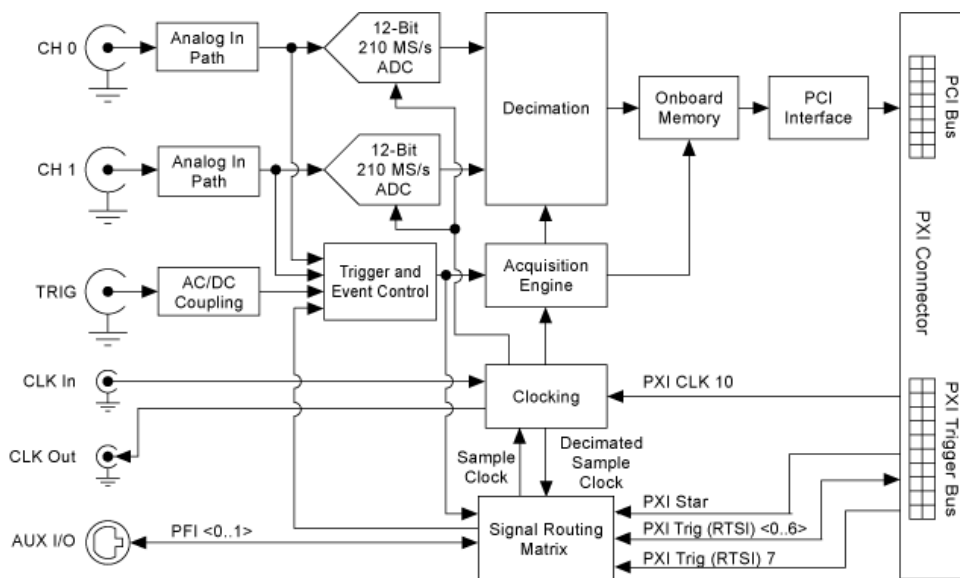
- PXI or PCI versions
- [NI-TClk synchronization](#)

Related Topics

[Features Supported by Device](#)

NI PXI-5124 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5124.



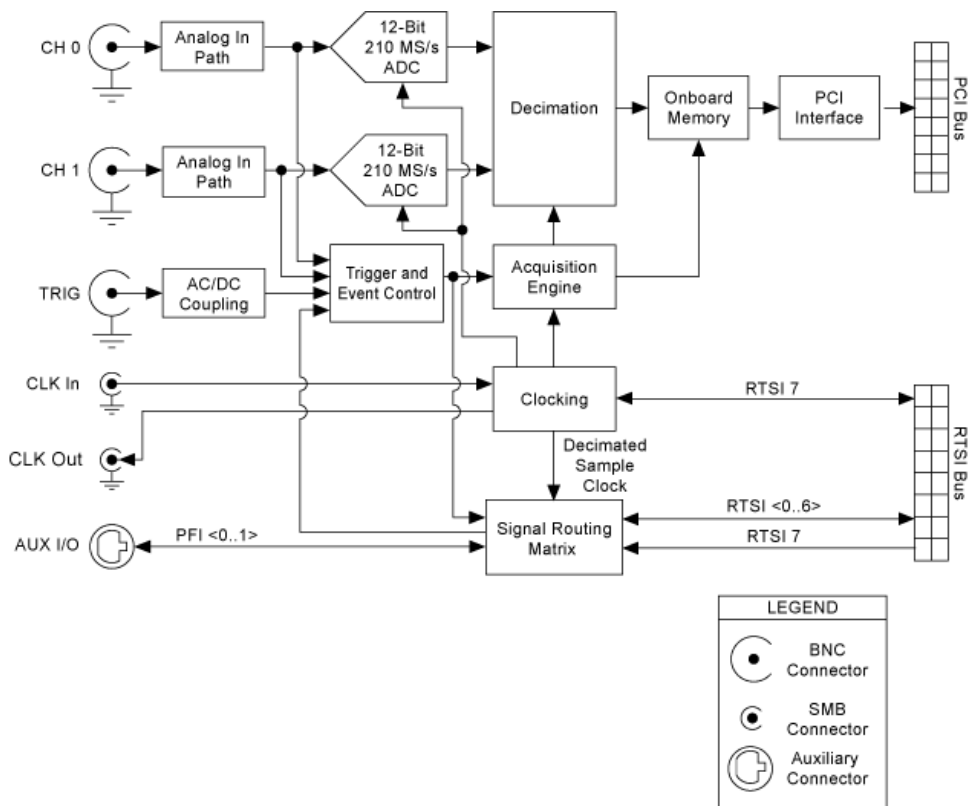
LEGEND	
	BNC Connector
	SMB Connector
	Auxiliary Connector

Related Topic

[NI PCI-5124 Block Diagram](#)

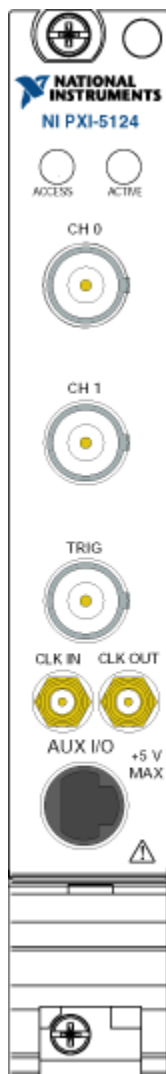
NI PCI-5124 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5124.



NI PXI-5124 Front Panel

The following figure shows the front panel of the NI PXI-5124. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI PXI-5124 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a trigger.
Green	The module has received a trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—the module is unable to detect the external sample clock. ▪ Overheating error—the module has detected an overheating error. Refer to Thermal Shutdown for more information. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>

Connectors

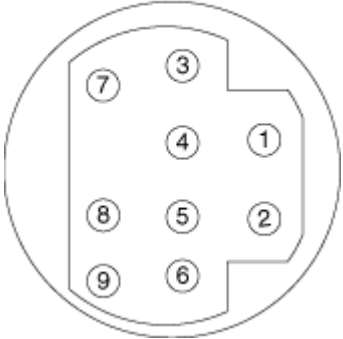
The NI PXI-5124 has the following six connectors on the front panel.


Connector	Description	Function
-----------	-------------	----------

CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

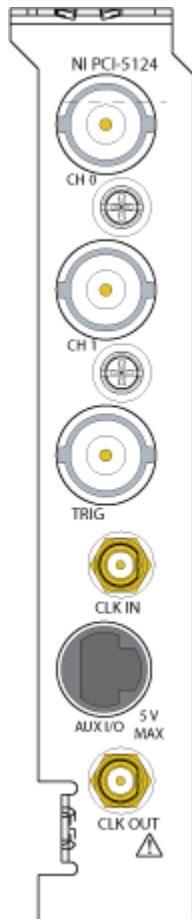
PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PCI-5124 Front Panel

The following figure shows the front panel of the NI PCI-5124. Descriptions of the connectors are shown below.



Connectors

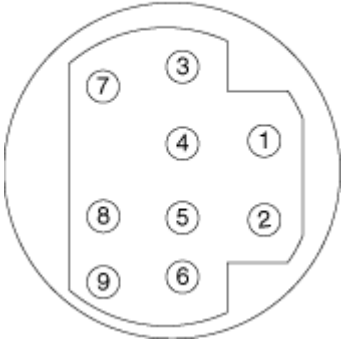
The NI PCI-5124 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized

CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0



Note Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over

temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.

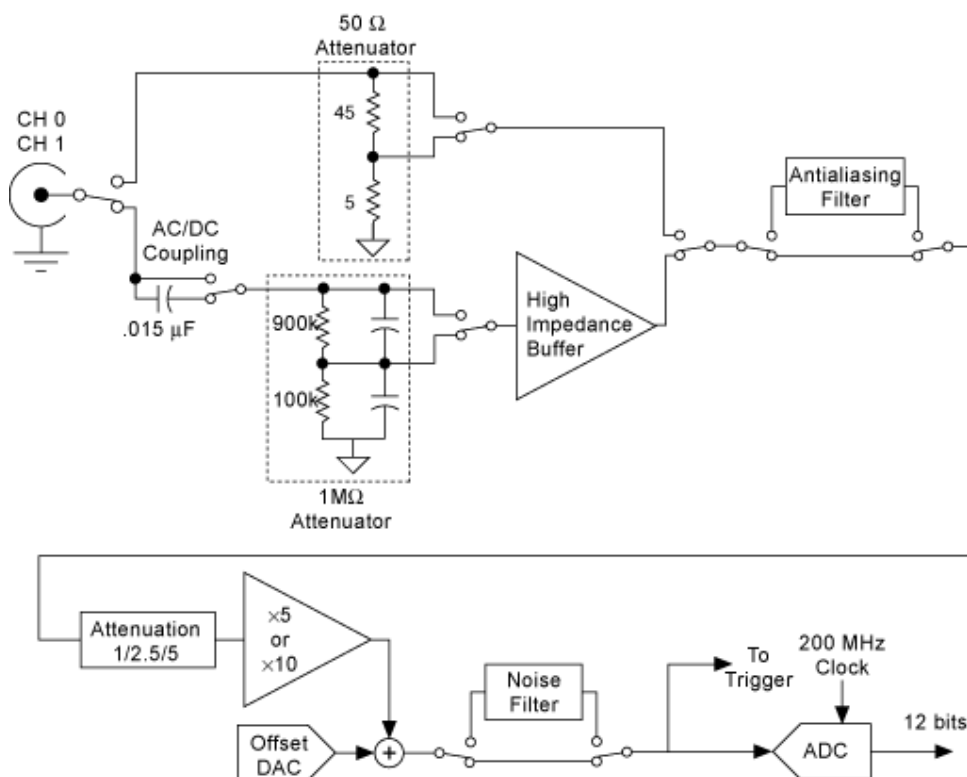
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

Features Supported by SMC-Based Devices and USB Devices

NI 5124 Input Signal Conditioning

The NI 5124 provides two independent digitizer input channel signal conditioning paths. Each path provides you with a choice of 50 Ω input impedance or 1 M Ω input impedance, as shown in the following diagram.



The 50 Ω path is optimized for flatness, distortion, dynamic range, and noise because the active circuitry in the signal path is minimized. In some applications, this optimization is the primary consideration. High-performance communications measurements can take advantage of the large [SFDR](#) and 12-bit dynamic range of the NI 5124.

The 1 M Ω path provides the traditional oscilloscope input impedance and loading characteristics for numerous general-purpose measurement applications.



Note The ground on the device inputs is connected to the chassis ground.

NI 5122/5124/5142 Input Ranges

As shown in the following table, the 1 M Ω path allows up to 20 V_{pk-pk} whereas the 50 Ω path is limited to 10 V_{pk-pk}.

50 Ω Input Path	1 M Ω Input Path
0.2 V _{pk-pk}	0.2 V _{pk-pk}
0.4 V _{pk-pk}	0.4 V _{pk-pk}
1 V _{pk-pk}	1 V _{pk-pk}
2 V _{pk-pk}	2 V _{pk-pk}
4 V _{pk-pk}	4 V _{pk-pk}
10 V _{pk-pk}	10 V _{pk-pk}
—	20 V _{pk-pk}



Note Except for the highest ranges (10 V for 50 Ω , 20 V for 1 M Ω), you can set the [vertical offset](#) on the NI 5122/5124/5142 to effectively create a unipolar range.

NI 5122/5124/5142 Input Impedance

You can set the NI 5122/5124/5142/5152 analog input impedance to either 50 Ω or 1 M Ω . The 50 Ω path provides slightly better noise and distortion performance because the high impedance buffer for the 1 M Ω path is bypassed. The 1 M Ω path is required in applications that require minimal loading or that require using a standard 10:1 oscilloscope probe.

Protection

The 50 Ω inputs of the NI 5122/5124/5142 are protected by a thermal disconnect circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω .

NI 5114/5122/5124/5142 AC/DC/GND Coupling

You can select AC, DC, or GND [input coupling](#) for the 1 M Ω input path, and DC or GND input coupling for the 50 Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit, or if you are using the 50 Ω path.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5114/5122/5124/5142 Vertical Offset

On NI 5114/5122/5124/5142 devices, you can set a hardware-applied vertical offset up to one-half of the given range, which effectively creates a unipolar range. For example, if you set the vertical offset to -1 V for the 2 V range, you can measure input voltages between 0 and -2 V.

 Note Vertical offset is not available on the 20 V range for 1 M Ω , or the 10 V range for 50 Ω .

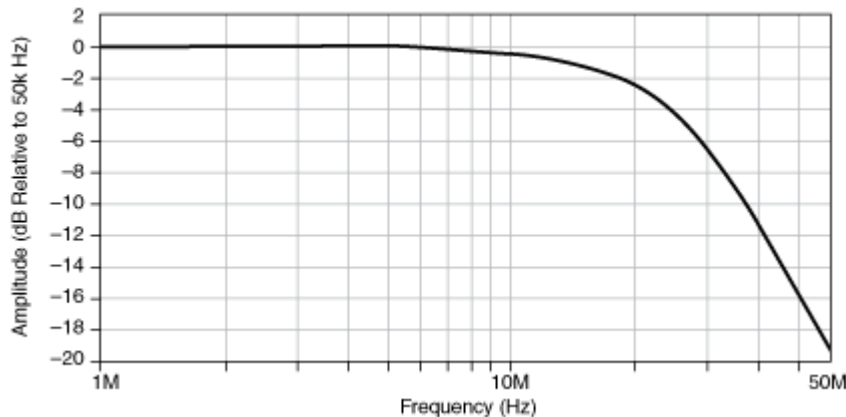
NI 5124 Filters

The NI 5124 offers two filters that limit the bandwidth of the signal path—a 20 MHz noise filter and a 60 MHz antialias filter.

 Note To change the filter type, call [niScope Configure Chan Characteristics VI](#) or the [niScope ConfigureChanCharacteristics](#) function and set the Max Input Frequency parameter to the appropriate value.

Noise Filter

A 20 MHz input filter is available through both the 1 M Ω and 50 Ω signal paths. This filter is intended to reduce noise when the signal content is 20 MHz or less. A typical frequency response of the noise filter is shown in the following figure.

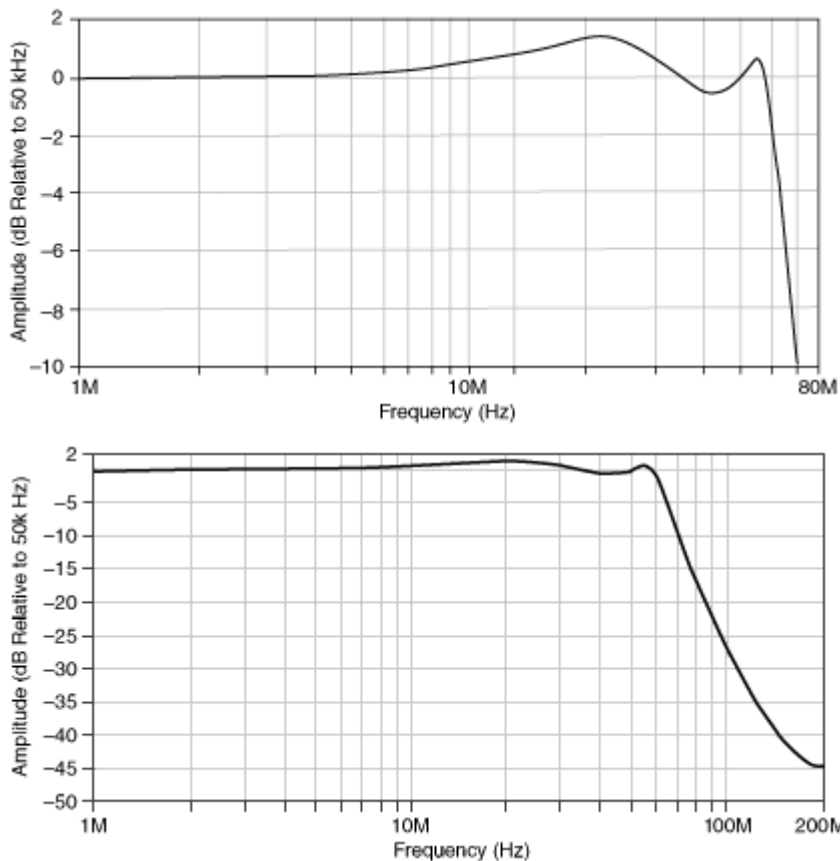


The noise filter provides over 20 dB of attenuation at 100 MHz, which is useful for rejecting out of band or background noise in lower frequency applications. An example application would be characterization of video waveforms in the presence of IF noise. Because most of the signal content is below the 8 MHz range, the 20 MHz noise filter is ideal. Many standard noise tests specify a 20 MHz noise bandwidth.

Antialias Filter

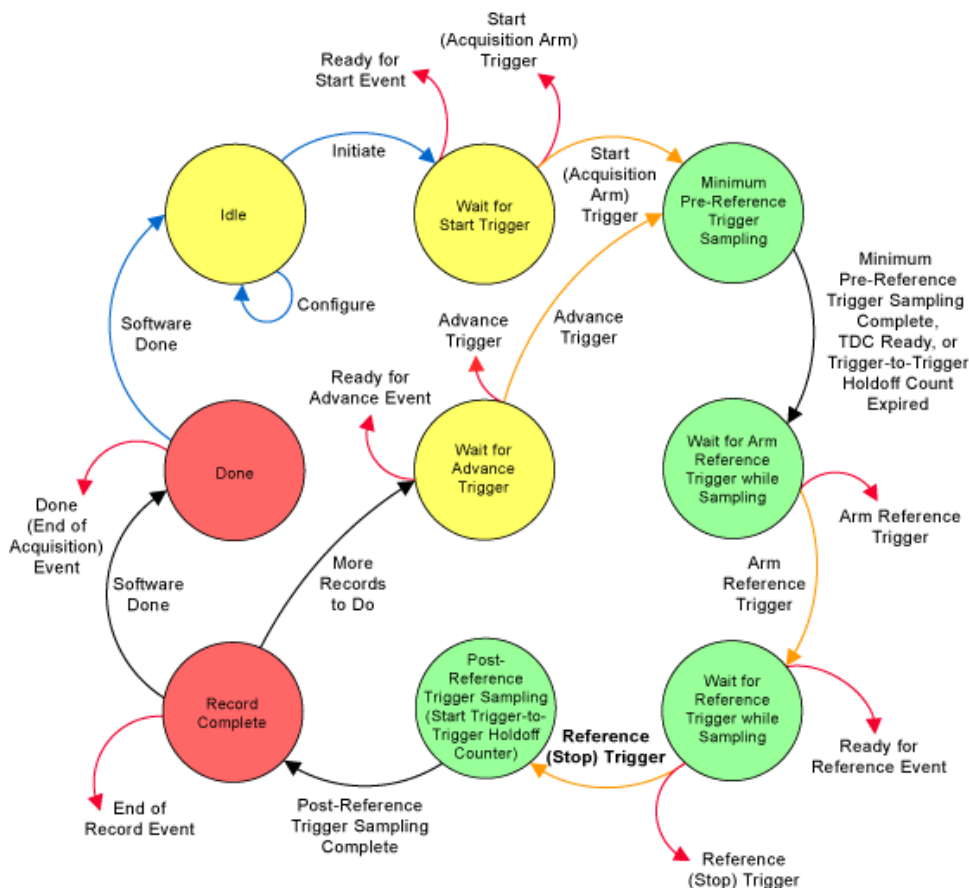
The NI 5124 provides another bandwidth-limiting filter called an antialias filter. Although this filter can be used to limit out of band noise in many applications, its primary purpose is to minimize [aliasing](#) effects caused by signals greater than [Nyquist](#) applied to the 200 MS/s ADC. Refer to the **NI PXI/PCI-5124 Specifications** for more information on this filter.

A typical response with the antialias filter is shown in the following two figures.



SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

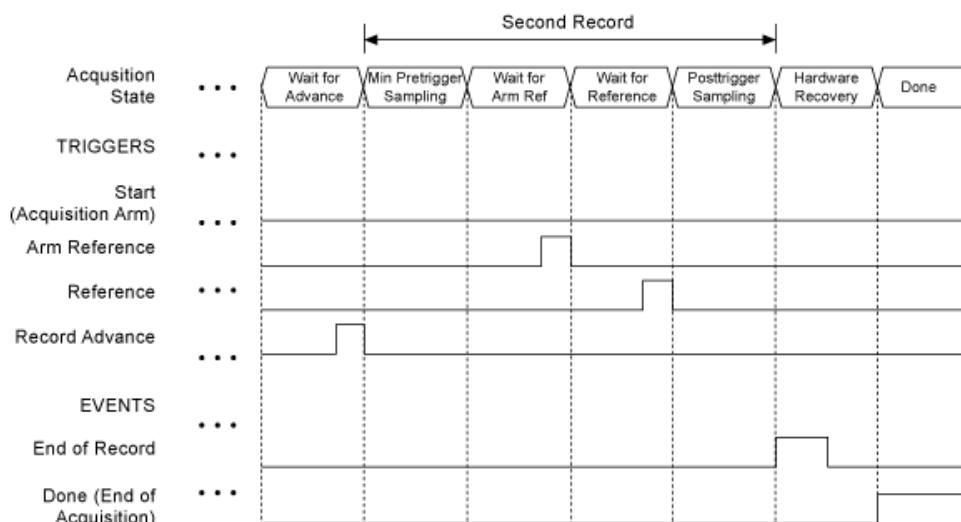
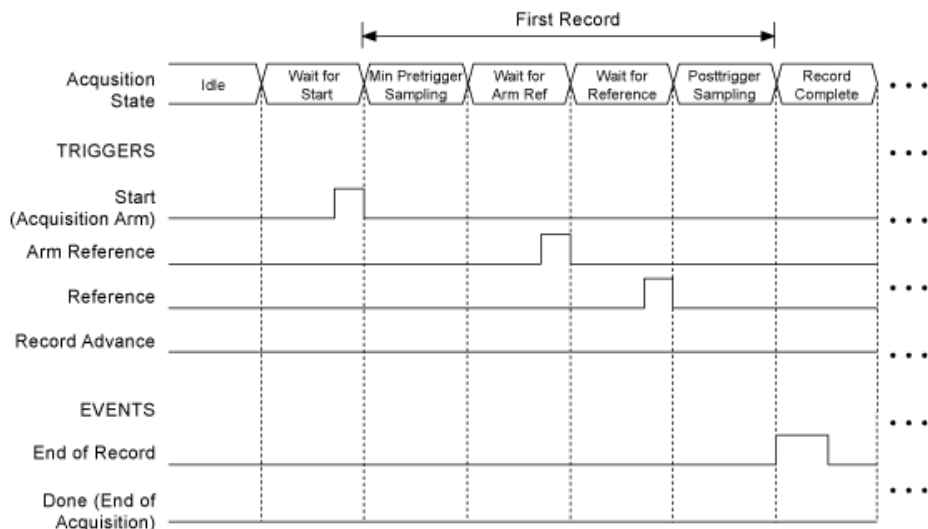
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)


SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.

 **Note** The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5122/5124/5142 and NI PXIe-5122 Routing Matrix

The following table shows the signals available for export from the NI PXI-5122/5124/5142 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)
Exported Clocks			
Reference Clock (External)	√	√	√
Sample Clock (Full Rate)	—	√	—
Sample Clock (Decimated Rate)	√	√	√
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

NI PCI-5122/5124/5142 Routing Matrix

The following table shows the signals available for export from the NI PCI-5122/5124/5142, and the lines to which they can be routed.

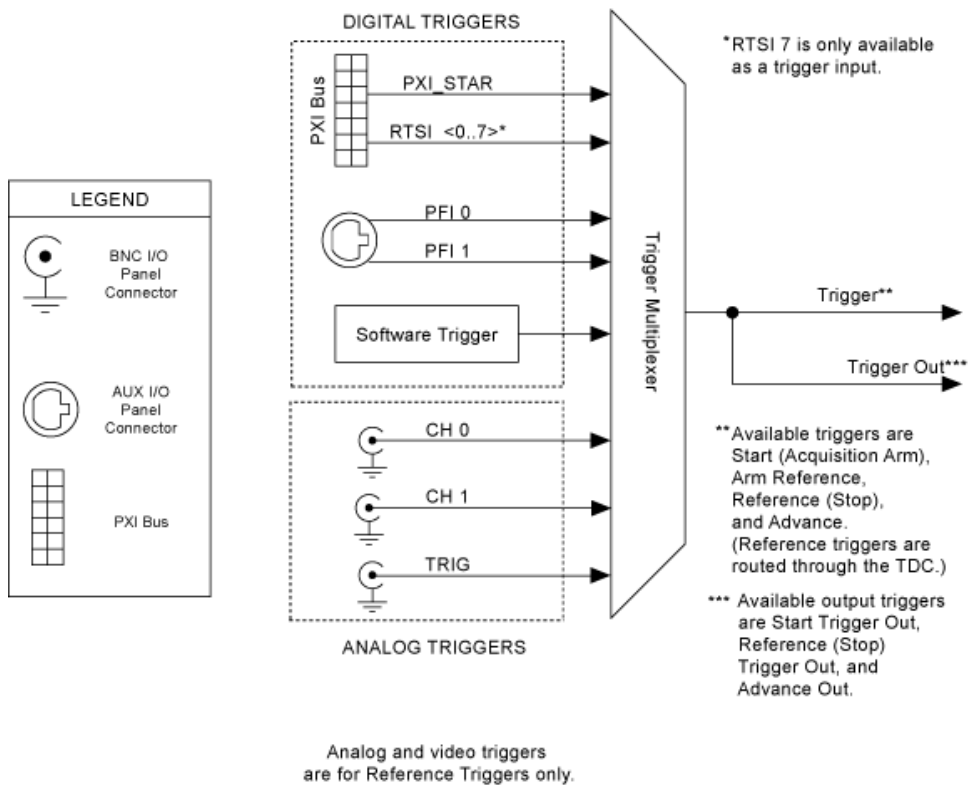
Source	Destination			
	RTSI <0..6> (RTSI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)	RTSI 7
Exported Clocks				
Reference Clock (External)	—	√	√	—
Reference Clock (Internal)	—	√	√	√
Sample Clock (Full Rate)	—	√	—	—
Sample Clock (Decimated Rate)	√	√	√	—
Triggers				
Acquisition Arm (Start) Trigger	√	—	√	—
Reference (Stop) Trigger	√	—	√	—
Events				
End of Record Event	√	—	√	—
End of Acquisition Event	√	—	√	—
Ready for Start Event	√	—	√	—
Ready for Reference Event	√	—	√	—
Ready for Advance Event	√	—	√	—

NI 5114/5122/5124/5142 Analog Trigger Types

The NI 5114/5122/5124/5142 supports the following analog trigger types: [edge](#), [hysteresis](#), [window](#), and [video](#) triggers.

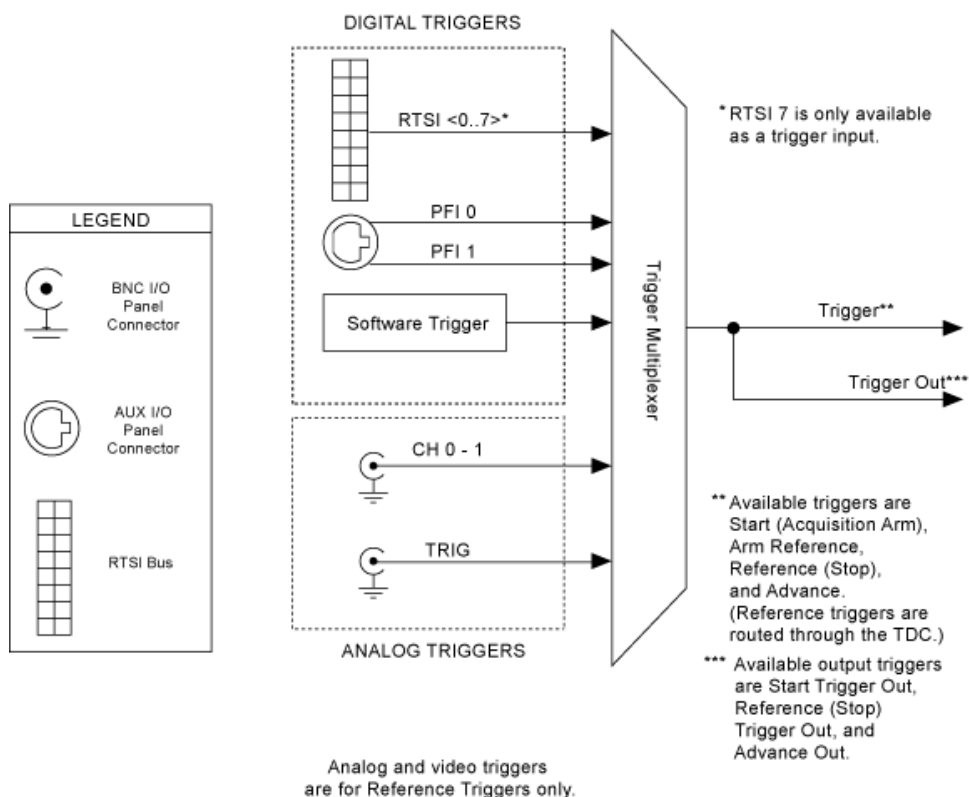
NI PXI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5114/5122/5124.



NI PCI-5114/5122/5124 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5114/5122/5124.



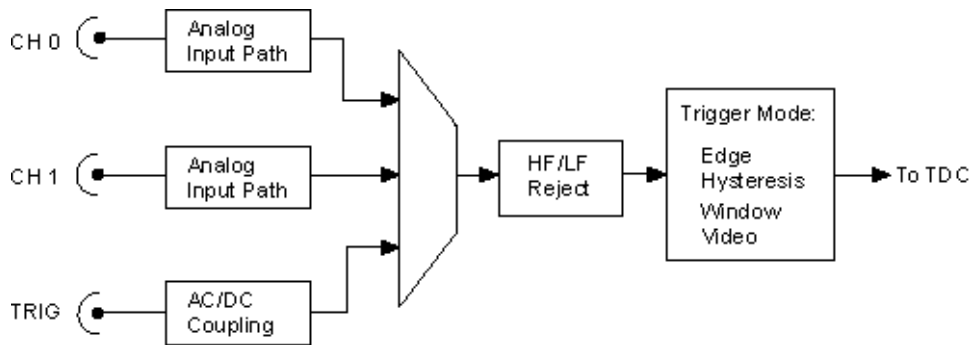
NI 5114/5122/5124/5142 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5114/5122/5124/5142 provides flexible, high-precision, low-jitter triggering features.

The NI 5114/5122/5124/5142 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5114/5122/5124/5142.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including

overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI 5122/5124/5142/5622/5922 Trigger Delay

[Trigger delay](#), which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

Max trigger delay in seconds = $[(2^{34} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock. The maximum trigger delay also changes based on the binary sample width. When binary sample width is set to 8, the maximum trigger delay is doubled. When binary sample width is set to 32, the maximum trigger delay is halved.

NI PXI-5124 Clocking

The clock circuitry on the NI PXI-5124 offers versatile clocking options with its ability to use either the internal 200 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PXI-5124 to phase lock the internal 200 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXI-5124.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 200 MHz clock. In these cases the NI PXI-5124 can accept an external sample clock. External clocking also provides a method to synchronize the NI PXI-5124 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PXI-5124 from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the **NI PXI/PCI-5124 Specifications** for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the **NI PXI/PCI-5124 Specifications** for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The [reference clock](#) is used in the NI PXI-5124 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The NI PXI-5124 can accept a reference clock from its front panel (CLK IN) as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PXI-5124 reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 200 MS/s when using the internal sample clock, the NI PXI-5124 decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PXI-5124 only stores one out of every four samples it receives from the ADCs to onboard memory. The NI PXI-5124 can sample at rates of $200/n$ MS/s, where n is an integer value between 1 and 2^{16} . Because the NI PXI-5124 does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When

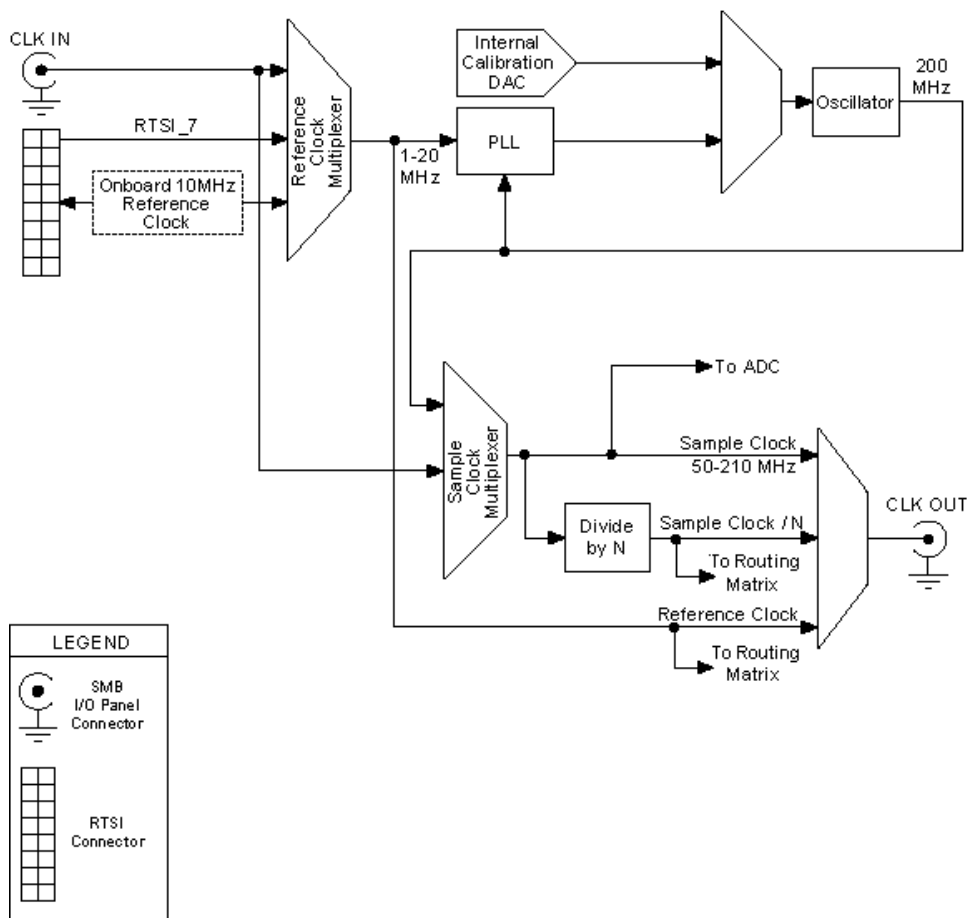
sampling at the maximum rate, a free-running version of the actual sample clock is exported to the front panel of the NI PXI-5124. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [Routing Matrix](#).

NI PCI-5124 Clocking

The clock circuitry on the NI PCI-5124 offers versatile clocking options with its ability to use either the internal 200 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PCI-5124 to phase lock the internal 200 MHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5124.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PCI-5124 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5124 has an onboard voltage controlled crystal oscillator (VCXO) running at 200 MHz. When using the onboard 200 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 200 MHz frequency of the VCXO. In PLL mode, the NI PCI-5124 phase locks its 200 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the NI PCI-5124 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 200 MHz clock. In these cases the NI PCI-5124 can accept an external sample clock. External clocking also provides a method to synchronize the NI PCI-5124 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PCI-5124 from the front panel connector. Refer to the **NI PXI/PCI-5124 Specifications** for external sample clock requirements.



Note When an external sample clock is used, the [TDC](#) is disabled. Refer to the **NI PXI/PCI-5124 Specifications** for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The reference clock is used in the NI PCI-5124 PLL circuit to synchronize the sample clock to the reference clock. The NI PCI-5124 can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 5 MHz to 20 MHz if it is provided to CLK IN or to RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PCI-5124 reference clock is None, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 200 MS/s when using the internal sample clock, the NI PCI-5124 decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PCI-5124 only stores one out of every four samples it receives from the ADCs to onboard memory. The NI PCI-5124 can sample at rates of $200/n$ MS/s, where n is an integer value between 1 and (2^{16}) . Because the NI PCI-5124 does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When

sampling at the maximum rate, a free-running version of the actual sample clock is exported to the front panel of the NI PCI-5124. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.

Exporting Reference Clock

If you are using the internal reference clock you can export that same reference clock to other NI PCI-5124 devices for synchronization purposes. For more information, refer to the section on synchronization. If you are using a reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [Routing Matrix](#).

NI 5124 Onboard Memory

The NI 5124 allocates at least 384 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is approximately 128 12-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the size of the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 128 samples. The available memory options are 4 MSamples, 16 MSamples, 128 MSamples, and 256 MSamples per channel.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5124 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition.

However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

NI 5122/5124/5142 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5122/5124/5142, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges, both input impedance paths (50 Ω and 1 MΩ), and all filter paths (enabled/disabled).
- Calibrates trigger level offset and gain.
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Compensates 1 MΩ input frequency flatness, then compensates input capacitance so that it is equal regardless of selected attenuator range.
- Takes approximately 2 minutes to complete.

Using the example as a comparison, the NI 5122/5124/5142, using self-calibration and a 5 V input, will have an error at 48 °C of

$$5 \text{ V}(0.65\% \text{ of input} + 8 \text{ mV}) = 40.5 \text{ mV error}$$

This result compares favorably to the 180 mV error for a typical digitizer as described in the example without self-calibration.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-

calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5122/5124/5142 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

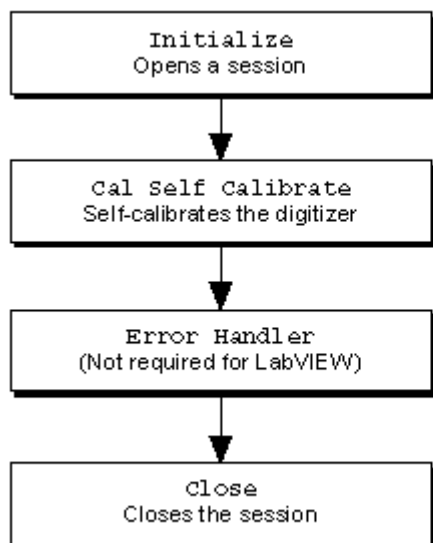
Input Connections During Self-Calibration

The NI 5122/5124/5142 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger level Trigger timing AC flatness Input capacitance	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy Optimizes performance with external 10:1 probes
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

NI USB-5132/5133

NI 5132/5133 high-speed digitizers are bus powered, 2-channel USB devices that are ideal for quickly obtaining time-domain measurements. The bus powered USB connection makes the digitizers highly mobile and thus a great solution for field engineers or for applications where individual measurements must be made at multiple systems.

Features

The NI 5132/5133 has the following features:

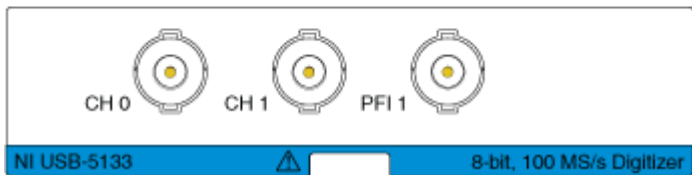
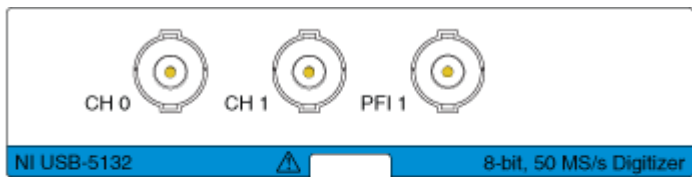
- 2 input channels
- 8-bit vertical resolution
- Real-time sample rate
 - NI 5132: 50 MS/s
 - NI 5133: 100 MS/s
- 50 MHz bandwidth (excluding 40 mV_{pk-pk} Range)
- AC/DC/GND coupling
- Analog and digital triggering
- Fully bus powered
- External clock

Related Topics

[Features Supported by Device](#)

NI USB-5132/5133 Front Panels

The following figures show the front panels of the NI USB-5132/5133. Descriptions of the connectors are shown below.



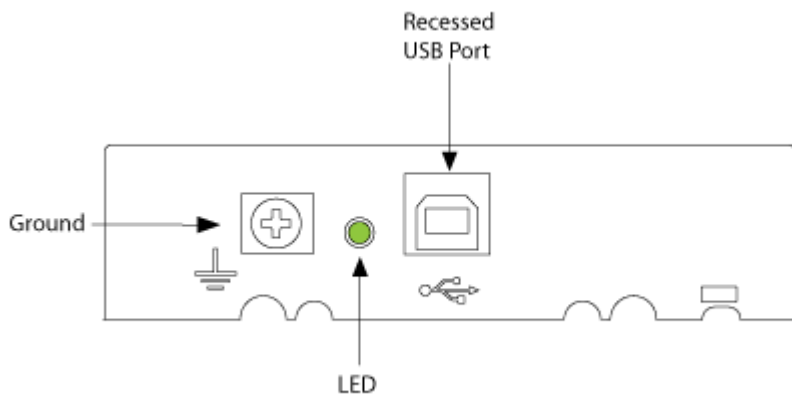
Connectors

The NI 5132/5133 has the following connectors on the front panel.

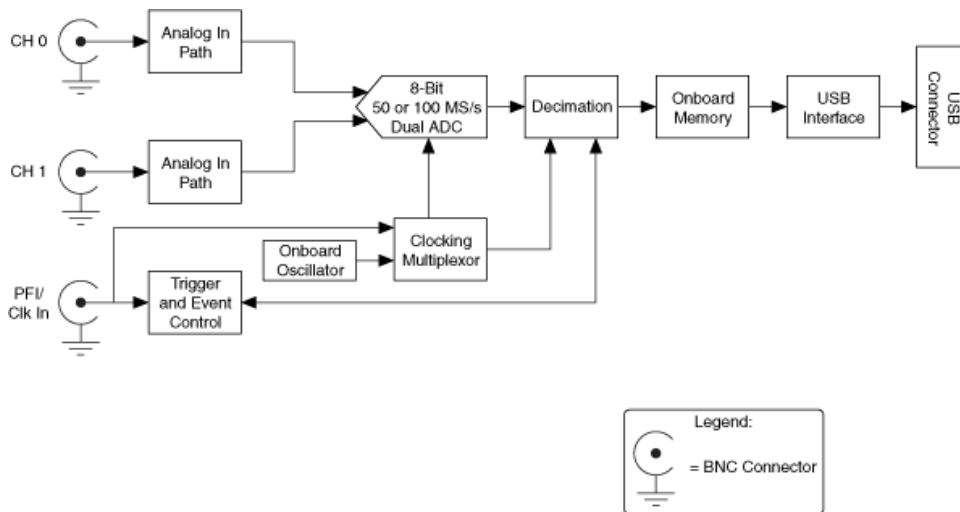
Connector	Description	Function
CH 0	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
PFI 1	Standard BNC connector	Multipurpose PFI line for trigger in/out, external clock in

NI USB-5132/5133 Back Panel

The following figure shows the back panel of the NI USB-5132/5133.

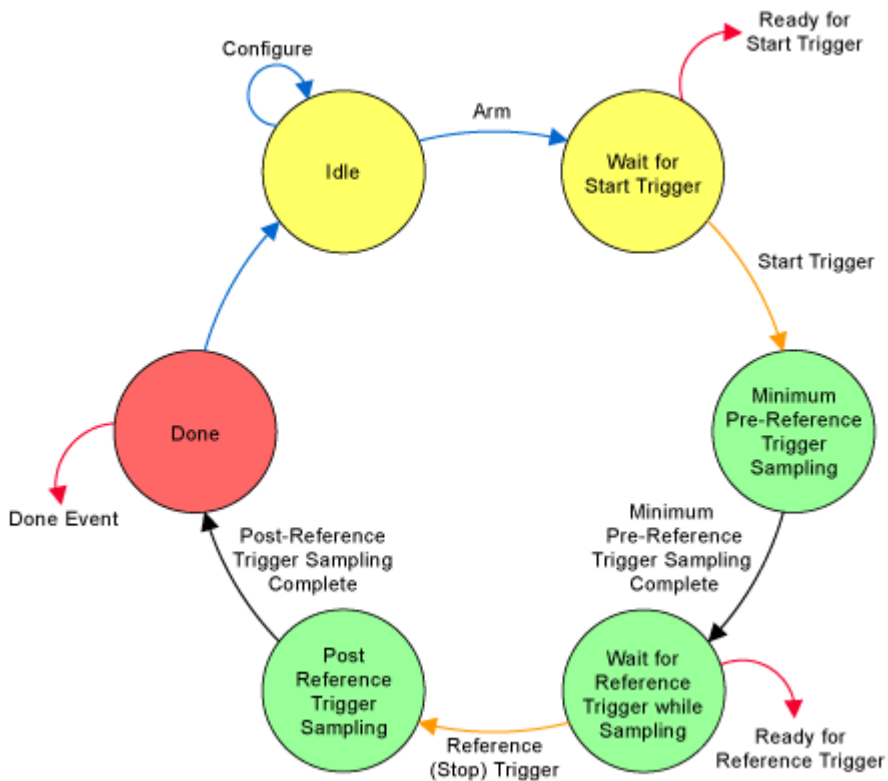


NI USB-5132/5133 Block Diagram



NI USB-5132/5133 Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for the NI USB-5132/5133.



ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI USB-5132/5133 can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module transitions into this state by receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source. While in this state, the module samples according to the session attributes configured. The module remains in this state until the minimum Pre-Reference Trigger sampling completes. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. When the minimum Pre-Reference Trigger sampling completes, the module transitions out of this state on the next clock cycle.

Wait for Reference—After the module has acquired the minimum pre-reference trigger samples, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. The module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Done—After the module completes the record, it transitions into this state. Upon entering this state, the module outputs the End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

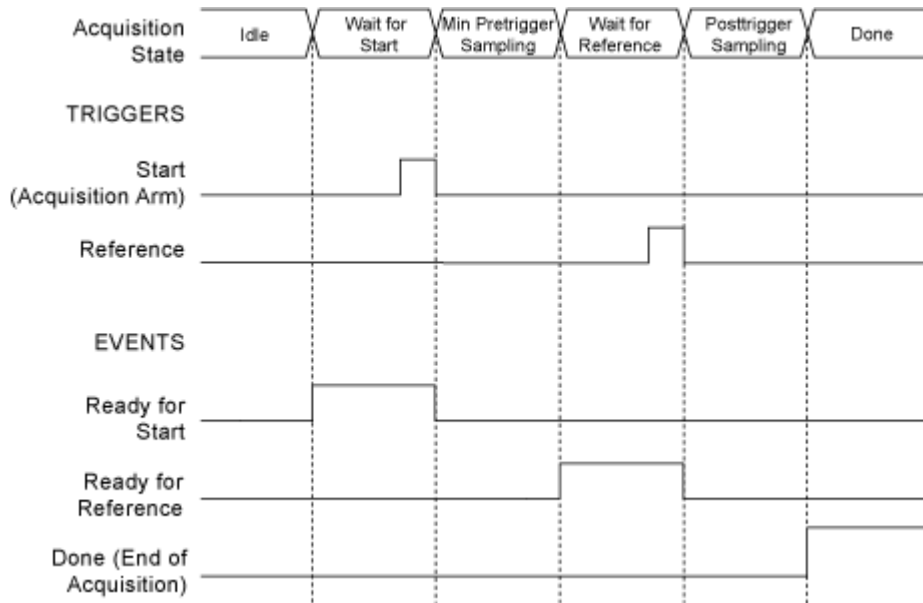
[Triggering](#)

[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

NI USB-5132/5133 Timing Diagram

The NI 5132/5133 supports single-record acquisitions only. The following timing diagram illustrates how NI 5132/5133 react to the user-configurable input triggers during acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state.



Note The trigger signals in the timing diagram assume active high level triggers.

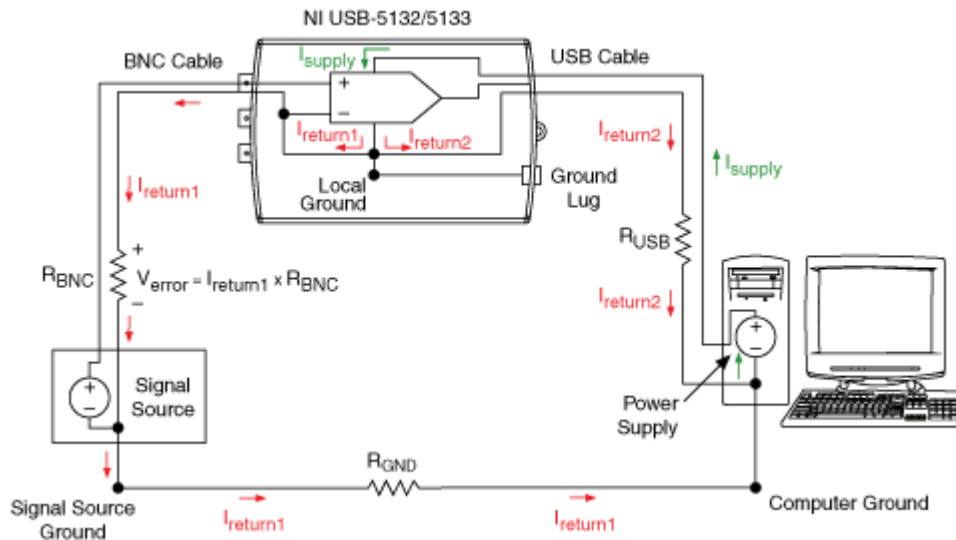
The exportable event (End of Acquisition) is also shown in the timing diagram. The End of Acquisition Event asserts when the record has been completed, but it does not assert if the acquisition is aborted or times out.

Ground Loop Noise in the NI USB-5132/5133

Some measurement setups can induce ground loop noise in measurements taken with the NI 5132/5133. This noise can appear as an undesired DC offset or as a stray low-frequency signal. This section describes methods you can use to mitigate ground-loop-related noise.

Ground-Loop-Related DC Offset

When a ground loop exists, a DC offset can appear in a measurement due to power supply return currents, as shown in the following figure.



The USB, BNC, and ground lug connectors all share the same ground inside the NI 5132/5133, thus power supply return currents to the host computer can flow out of all three connections. Ideally, all of the supply current returns through the USB cable (the current labeled $I_{return2}$) and none returns through the BNC cable (the current labeled $I_{return1}$). In practice, this does not happen because R_{USB} is not a perfect short and R_{GND} is finite when a ground loop exists. As shown in the figure, the unwanted voltage drop along the BNC cable, V_{error} , is in series with the measured voltage and represents a DC offset error. By Ohms law, V_{error} is as follows:

$$V_{error} = I_{return1} \times R_{BNC}$$

Also, given the three ground resistances depicted in the figure, you can predict the magnitude of $I_{return1}$ with respect to the total supply current, I_{supply} , using the current division formula as follows:

$$I_{return1} = (R_{USB} / (R_{GND} + R_{USB} + R_{BNC})) \times I_{supply}$$

Substitute the second equation into the first:

$$V_{error} = ((R_{USB} \times R_{BNC}) / (R_{GND} + R_{USB} + R_{BNC})) \times I_{supply}$$

As the previous formula shows, you can reduce V_{error} by increasing R_{GND} , reducing R_{USB} , or reducing R_{BNC} .

How to Reduce Ground-Loop-Related Offset Error

To reduce V_{error} , use the following methods:

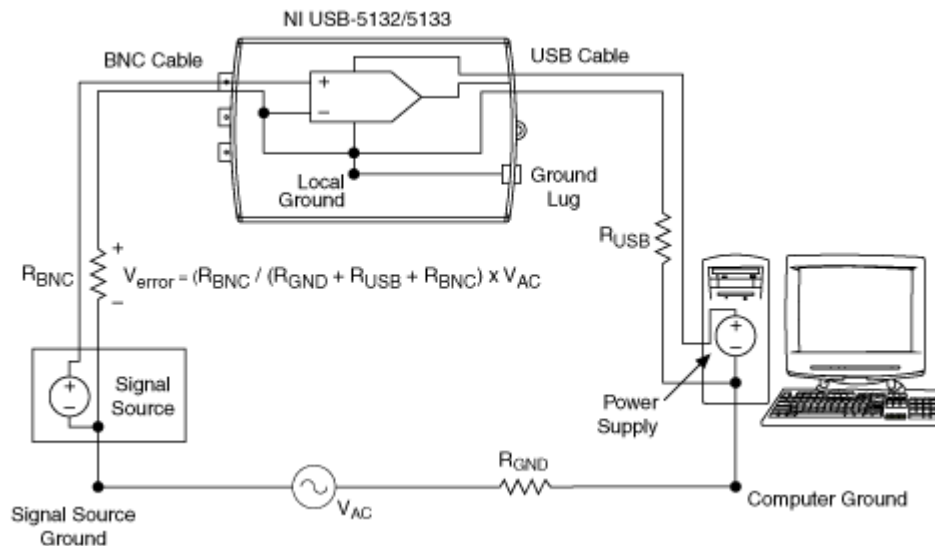
- Increase R_{GND} by disconnecting the ground between the measurement source and the computer ground (in other words, break the ground loop). If this is not possible, increase R_{GND} by some other method. Note that increasing R_{GND} to only 1 Ohm usually eliminates any measurable DC offset.
- Reduce R_{USB} by using a higher quality or shorter USB cable. Alternatively, install a heavy ground wire between the NI 5132/5133 ground lug and the host computer ground (thus effectively reducing the value of R_{USB}).
- Reduce R_{BNC} by using a shorter BNC cable. Alternatively, install a heavy ground wire between the NI 5132/5133 ground lug and the signal source ground (thus effectively reducing the value of R_{BNC}).



Note You can also perform any combination of these methods to further reduce V_{error} .

Ground-Loop-Related AC Noise

An AC noise source existing in a ground loop can appear in NI 5132/5133 measurements, as shown in the following figure.



A loop is formed by the BNC cable, the NI 5132/5133, the USB cable, the computer, the ground interconnection between computer and signal source, and the signal

source. An AC noise source, V_{AC} , is depicted in series with the loop and causes the generation of an error voltage V_{error} in series with the NI 5132/5133 measurement input. Because the loop forms a voltage divider, you can write an equation for V_{error} in terms of V_{AC} and the three resistances around the loop as follows:

$$V_{error} = (R_{BNC} / (R_{GND} + R_{USB} + R_{BNC})) \times V_{AC}$$

As the previous formula shows, you can reduce V_{error} by increasing R_{GND} , reducing R_{BNC} , or reducing V_{AC} .

How to Reduce Ground-Loop-Related AC Noise

To reduce V_{error} , use the following methods:

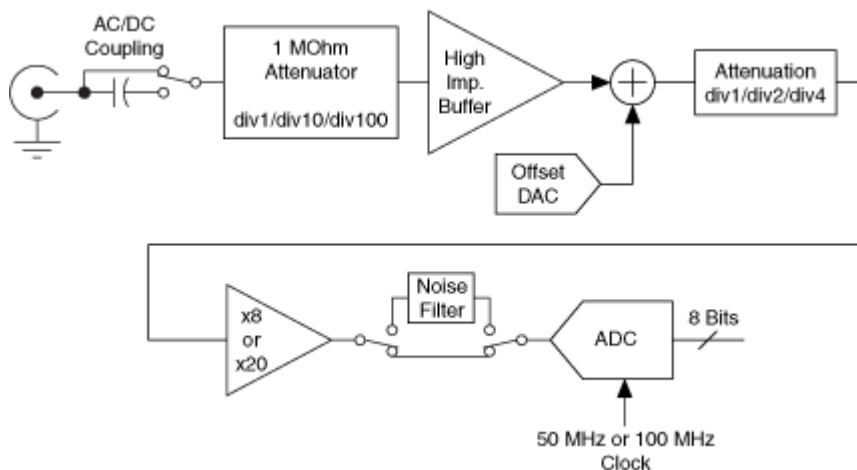
- Increase R_{GND} by disconnecting the ground between the measurement source and the computer ground (break the ground loop). If breaking the ground loop is not possible, a modest increase in the value of R_{GND} can help. For example, if the total resistance around a ground loop is initially 0.1 Ohms, increasing the value of R_{GND} to 1 Ohm decreases ground loop noise by 20 dB.
- Reduce R_{BNC} by using a shorter BNC cable. Alternatively, install a heavy ground wire between the NI 5132/5133 ground lug and the signal source ground, thus effectively reducing the value of R_{BNC} .
- Reduce V_{AC} . You can accomplish this goal in several ways. Ideally, the host computer and measurement source should share a ground interconnection that carries little current. Both should share the same power outlet or power strip, when possible. Alternatively, you can connect a heavy wire between computer ground and measurement source ground to shunt V_{AC} . The resistance of this wire would need to be much lower than R_{GND} . If the noise is induced in the loop magnetically, reduce the area of the loop or separate any cables in the loop from cables carrying large AC currents.



Note You can also perform any combination of these methods to further reduce V_{error} .

NI USB-5132/5133 Signal Conditioning

The following diagram shows the 1 M Ω input path of the NI 5132/5133.



The 1 M Ω path provides the traditional oscilloscope input impedance and loading characteristics for numerous general-purpose measurement applications.



Note The ground on the device inputs is connected to the chassis ground.

NI USB-5132/5133 Input Ranges

The following table shows the input ranges of the NI 5132/5133 for the 1 M Ω path.

1 M Ω Input Path
0.04 V _{pk-pk}
0.1 V _{pk-pk}
0.2 V _{pk-pk}
0.4 V _{pk-pk}
1 V _{pk-pk}
2 V _{pk-pk}
4 V _{pk-pk}
10 V _{pk-pk}
20 V _{pk-pk}

40 V_{pk-pk}

NI USB-5132/5133 Vertical Offset

The following table shows the valid vertical offset for each range on the NI 5132/5133.

Input Range V _{pk-pk}	Vertical Offset Range	Maximum Signal Swing Plus Offset Must Be Below
40	±30 V	±30 V
20	±30 V	±30 V
10	±30 V	±30 V
4	±4 V	±6 V
2	±4 V	±5 V
1	±4 V	±4.5 V
0.4	±0.4 V	±0.6 V
0.2	±0.4 V	±0.5 V
0.1	±0.4 V	±0.45 V
0.04	±0.4 V	±0.402 V

NI USB-5132/5133 Input Impedance

The input impedance of the NI 5132/5133 is 1 MΩ. This input impedance is required in applications that demand minimal loading or that necessitate using a standard 10:1 oscilloscope probe.

NI USB-5132/5133 AC/DC/GND Coupling

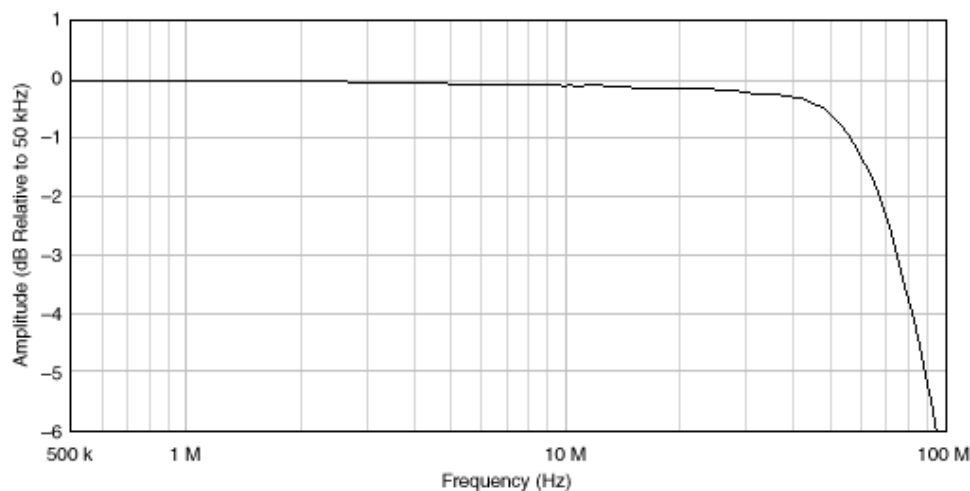
The NI 5132/5133 has software selectable AC, DC, and GND coupling. Select AC coupling if the input signal has a DC component that you want to reject, provided you are not concerned about low-frequency flatness. A DC input adjustment is available if you do not want to enable AC coupling.

Ground coupling internally connects the input to the local ground.

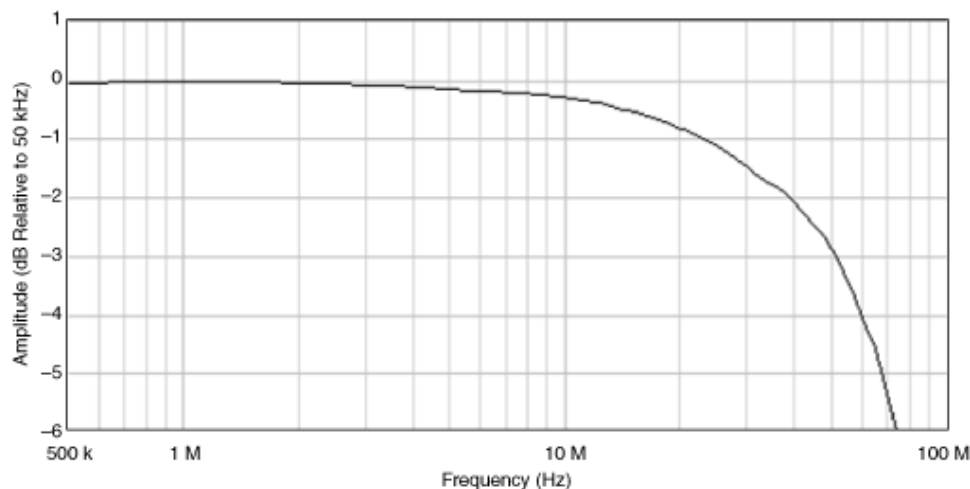
NI USB-5132/5133 Noise Filter

The NI 5132/5133 has a software selectable 20 MHz noise filter that limits the bandwidth of the signal path. The filter is intended to reduce noise when the input signal content is 20 MHz or less.

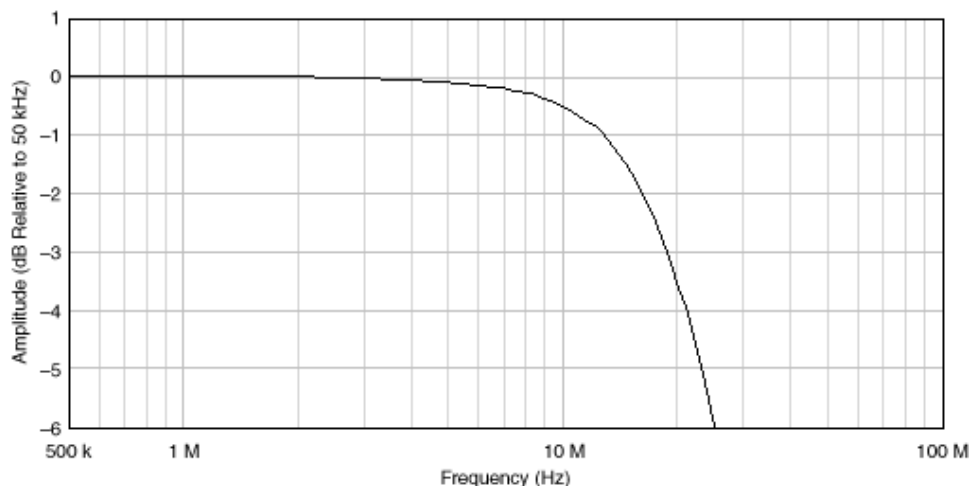
The following figure shows a typical frequency response at full bandwidth (excluding $40\text{ mV}_{\text{pk-pk}}$).



The following figure shows a typical frequency response at full bandwidth ($40\text{ mV}_{\text{pk-pk}}$ only)

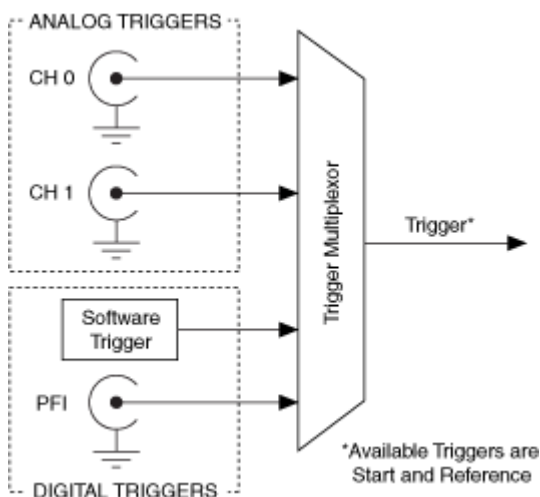


The following figure shows a typical frequency response with the 20 MHz noise filter.



NI USB-5132/5133 Trigger Sources

The following figure shows the trigger sources for NI USB-5132/5133 digitizers.



NI USB-5132/5133 Analog Trigger Paths

The NI 5132/5133 has two fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)

The analog trigger path for the NI 5132/5133 is the same as the [input signal path](#). Thus, analog triggers are subject to the same limitations as an input signal.

NI USB-5132/5133 Trigger Types

The NI 5132/5133 has six trigger types:

- Immediate
- Software
- Digital Edge (PFI 1)
- Analog Edge (CH 0, CH 1)
- Window (CH 0, CH 1)
- Hysteresis (CH 0, CH 1)



Note If a digital trigger is being supplied through the PFI line you cannot use an external clock.

NI USB-5132/5133 Trigger Delay

Trigger delay, which is specified in seconds, is achieved on the NI 5132/5133 by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

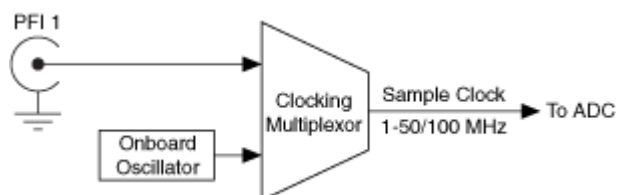
$$[(2^{32} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$



Note The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock.

NI USB-5132/5133 Clocking

The clock circuitry on the NI 5132/5133 offers versatile clocking options with its ability to use either the internal 50/100 MHz sample clock or to accept an external sample clock that you provide. The following diagram shows the clocking options of the NI 5132/5133.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI 5132/5133 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI 5132/5133 has an onboard crystal oscillator (XO) running at 50 MHz (NI 5132) or 100 MHz (NI 5133). This clock is sent to the ADC and to the input timing engine.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 50/100 MHz clock. In these cases the NI 5132/5133 can accept an external sample clock. External clocking also provides a method to synchronize the NI 5132/5133 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI 5132/5133 from PFI 1 on the front panel. Refer to the [NI 5132/5133 Specifications](#) document that shipped with the device for external sample clock requirements.

NI USB 5132/5133 Routing Matrix

The following table shows the signals available for export from the NI USB 5132/5133 and the lines to which they can be routed.

Source	Destination
	PFI 1
Triggers	
Acquisition Arm (Start) Trigger	√
Reference (Stop) Trigger	√
Events	
End of Acquisition Event	√
Ready for Start Event	√

Ready for Reference Event

√

NI USB-5132/5133 Onboard Memory

The total number of samples that can be stored depends on the acquisition memory size option. Refer to the hardware specifications document for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5133/5132 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.



Note To specify the posttrigger sample count, use the [Reference Position](#) property or the [NISCOPE_ATTR_REF_POSITION](#) attribute. This property/attribute specifies what percentage of the record should be pretrigger samples.

NI USB-5132/5133 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{--}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple instruments integrated together the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filter are among these factors.

If the ambient temperature is outside of the 18–28 °C range you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to achieve the specified accuracy outside of the range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration.

In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC), error is specified as:

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is:

$$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function. For the NI 5132/5133, this self-calibration capability yields the following benefits:

- Corrects for DC offset errors within the digitizer. This correction is done for all ranges and all filter paths (enabled/disabled).
- Takes approximately 10 seconds to complete.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy. When the two-year calibration interval expires, an external calibration is required.

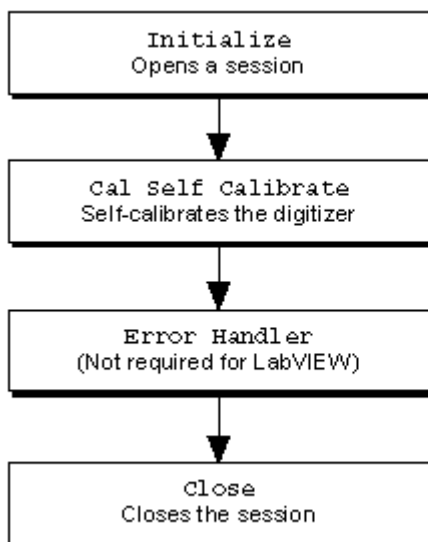
The NI 5132/5133 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

Input Connections During Self-Calibration

The NI 5132/5133 inputs are internally grounded during self-calibration. However, problems may occur if any signal is present at the input (especially high frequency AC signals). NI recommends completely disconnecting any signal during self calibration. If removing the signal source is not an option, it is important to ensure that the source is turned off and no signal is being put into the board.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at Start»All Programs»National Instruments»NI-SCOPE»Examples.

Summary of Calibration Options

A summary of the calibration options available for the NI 5132/5133 and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External Calibration	Gain AC Flatness User Input Offset	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset	90 days, or when temperature changes $>5^{\circ}\text{C}$	Calibrates out internal offset errors
No calibration	None, within 2 year calibration cycle or if temperature stays within $\pm 5^{\circ}\text{C}$	High accuracy not required outside of 5°C	If self-calibration is not used, derate the accuracy using the specified Temperature Coefficient

NI 5142

NI 5142 high-speed digitizers with onboard signal processing feature large dynamic range, wide analog bandwidth, and deep onboard acquisition memory for both time and frequency domain measurements. The NI 5142 can be used in conjunction with the NI 5600 downconverter (as the NI PXI-5661 RF signal analyzer) or as a standalone instrument.

Features

The NI 5142 has the following features:

- 2 channels, simultaneously sampled
- 14-bit vertical resolution
- 100 MS/s real-time sampling rate
- 2.0 GS/s random interleaved sampling (RIS) rate
- 100 MHz bandwidth
- 64 or 256 MB of memory per channel
- Onboard signal processing

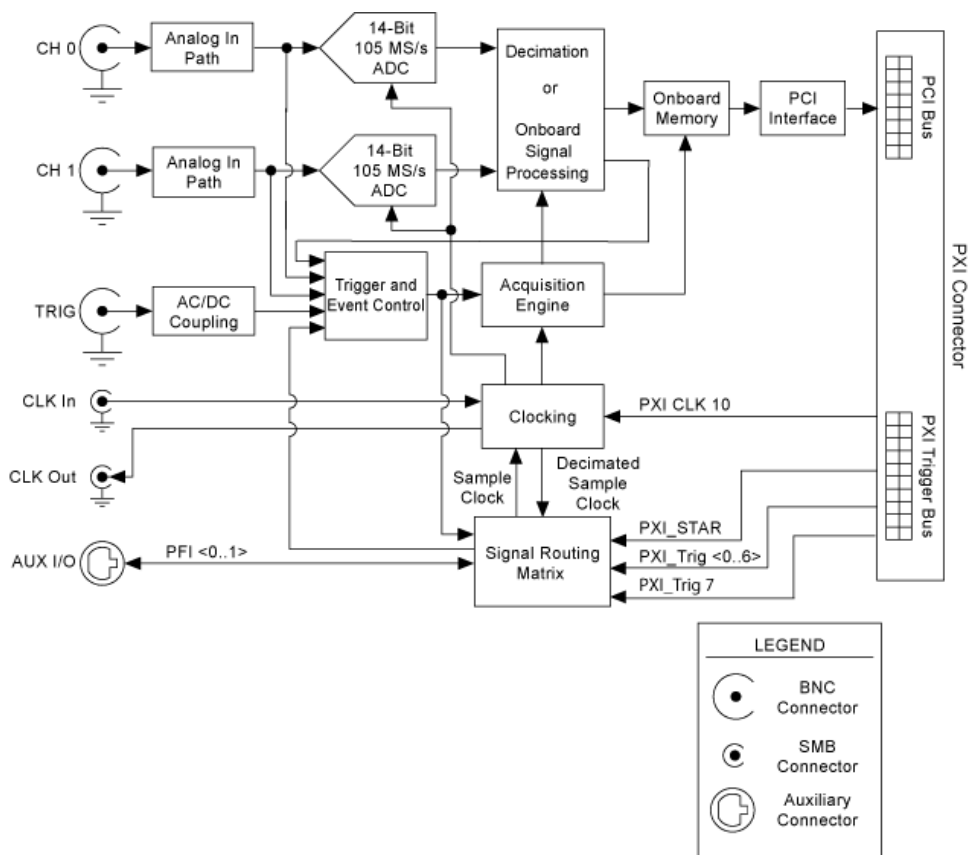
- [NI-TClk synchronization](#)
- PXI or PCI versions

Related Topics

[Features Supported by Device](#)

NI PXI-5142 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5142.

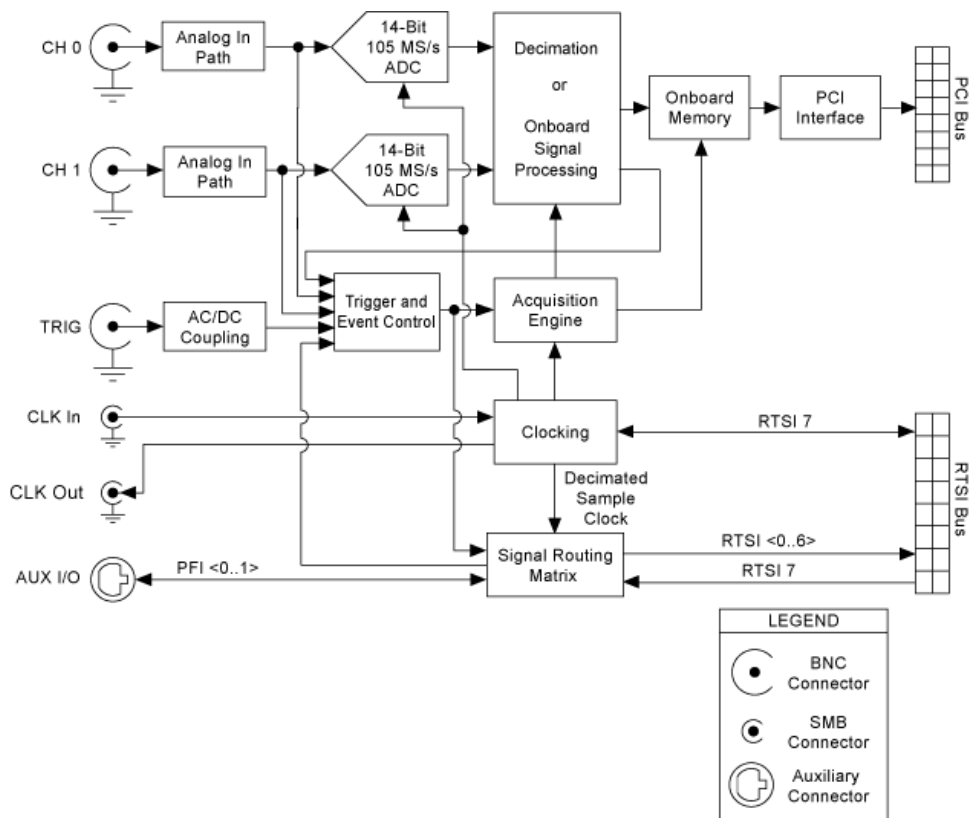


Related Topics

[Onboard Signal Processing](#)

NI PCI-5142 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5142.

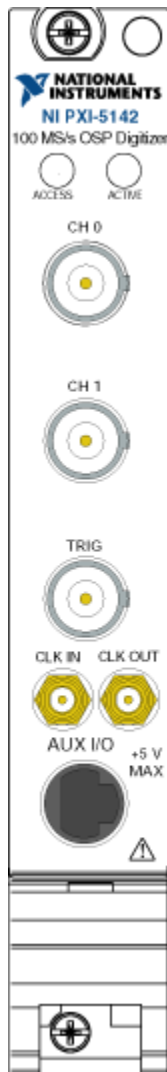


Related Topics

[Onboard Signal Processing](#)

NI PXI-5142 Front Panel

The following figure shows the front panel of the NI PXI-5142. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI PXI-5142 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—the module is unable to detect the external sample clock. ▪ Overheating error—the module has detected an overheating error. Refer to Thermal Shutdown for more information. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>

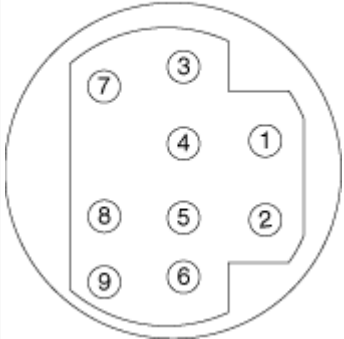
Connectors


The NI PXI-5142 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

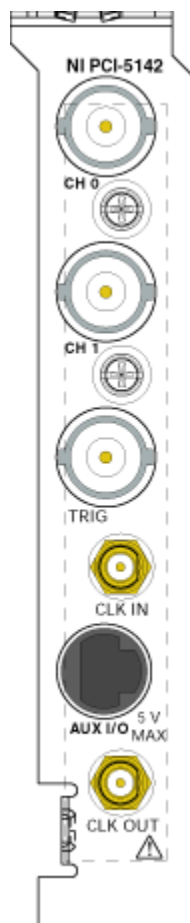
PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PCI-5142 Front Panel

The following figure shows the front panel of the NI PCI-5142. Descriptions of the LEDs and connectors are shown below.



Connectors

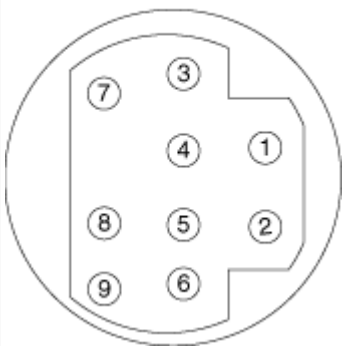
The NI PCI-5142 has the following six connectors on the front panel.


Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions

TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference or sample clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference or sample clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

PFI 0 and PFI 1 are accessible through the 9-pin connector (AUX I/O), shown in the following figure.

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

 Notes

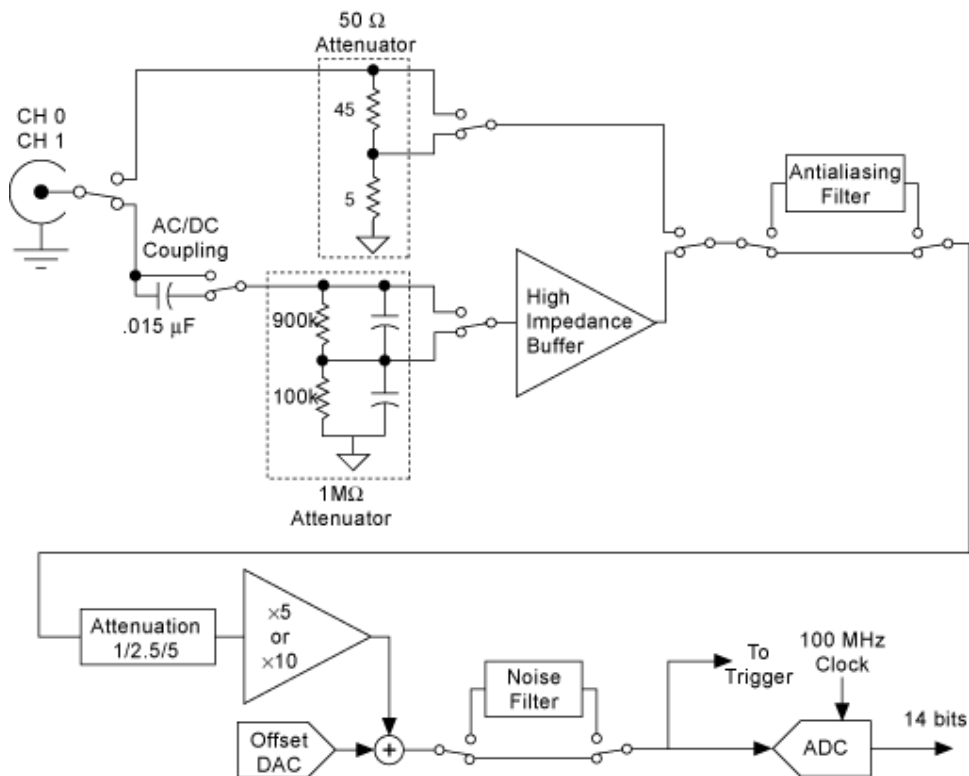
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)


NI 5122/5142 Input Signal Conditioning

The NI 5122/5142 provides two independent digitizer input channel signal conditioning paths. Each path provides you with a choice of 50 Ω input impedance or 1 M Ω input impedance, as shown in the following diagram.



The 50 Ω path is optimized for flatness, distortion, dynamic range, and noise because the active circuitry in the signal path is minimized. In some applications, this optimization is the primary consideration. High-performance communications measurements can take advantage of the large SFDR and 14-bit dynamic range of the NI 5122/5142.

The 1 MΩ path provides the traditional oscilloscope input impedance and loading characteristics for numerous general-purpose measurement applications.


 Note The ground on the device inputs is connected to the chassis ground.

NI 5122/5124/5142 Input Ranges

As shown in the following table, the 1 MΩ path allows up to 20 V_{pk-pk} whereas the 50 Ω path is limited to 10 V_{pk-pk}.

50Ω Input Path	1MΩ Input Path
0.2 V _{pk-pk}	0.2 V _{pk-pk}
0.4 V _{pk-pk}	0.4 V _{pk-pk}

1 V _{pk-pk}	1 V _{pk-pk}
2 V _{pk-pk}	2 V _{pk-pk}
4 V _{pk-pk}	4 V _{pk-pk}
10 V _{pk-pk}	10 V _{pk-pk}
—	20 V _{pk-pk}

 **Note** Except for the highest ranges (10 V for 50 Ω, 20 V for 1 MΩ), you can set the [vertical offset](#) on the NI 5122/5124/5142 to effectively create a unipolar range.

NI 5122/5124/5142 Input Impedance

You can set the NI 5122/5124/5142/5152 analog input impedance to either 50 Ω or 1 MΩ. The 50 Ω path provides slightly better noise and distortion performance because the high impedance buffer for the 1 MΩ path is bypassed. The 1 MΩ path is required in applications that require minimal loading or that require using a standard 10:1 oscilloscope probe.

Protection

The 50 Ω inputs of the NI 5122/5124/5142 are protected by a thermal disconnect circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω.

NI 5114/5122/5124/5142 AC/DC/GND Coupling

You can select AC, DC, or GND [input coupling](#) for the 1 MΩ input path, and DC or GND input coupling for the 50 Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit, or if you are using the 50 Ω path.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5114/5122/5124/5142 Vertical Offset

On NI 5114/5122/5124/5142 devices, you can set a hardware-applied vertical offset up to one-half of the given range, which effectively creates a unipolar range. For example, if you set the vertical offset to -1 V for the 2 V range, you can measure input voltages between 0 and -2 V.



Note Vertical offset is not available on the 20 V range for 1 M Ω , or the 10 V range for 50 Ω .

NI 5122/5142 Filters

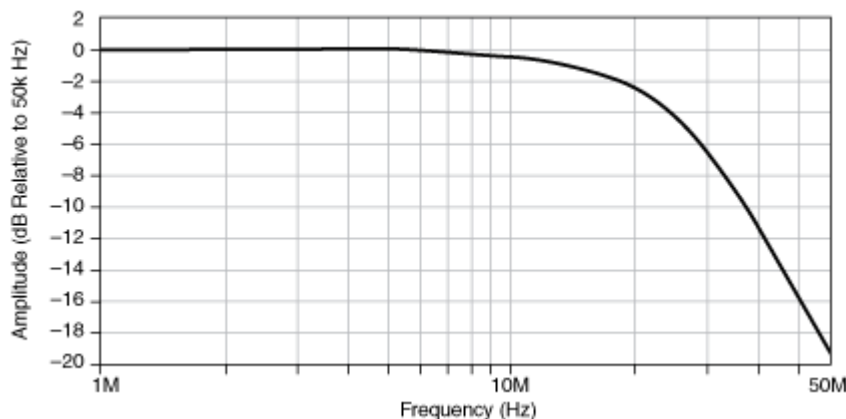
The NI 5122/5142 offers two filters that limit the bandwidth of the signal path—a 20 MHz noise filter and a 35 MHz antialias filter.



Note To change the filter type, call [niScope Configure Chan Characteristics VI](#) or the [niScope_ConfigureChanCharacteristics](#) function and set the Max Input Frequency parameter to the appropriate value.

Noise Filter

A 20 MHz input filter is available through both the 1 M Ω and 50 Ω signal paths. This filter is intended to reduce noise when the signal content is 20 MHz or less. A typical frequency response of the noise filter is shown in the following figure.

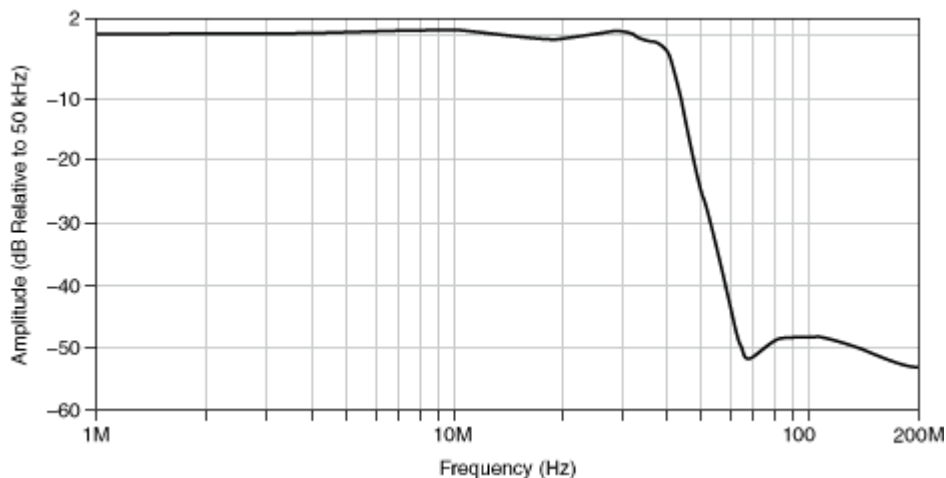
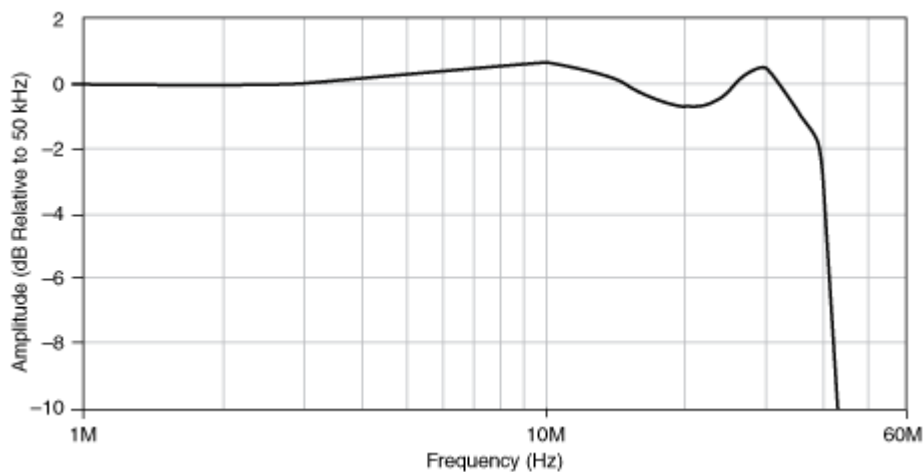


The noise filter provides over 20 dB of attenuation at 100 MHz, which is useful for rejecting out of band or background noise in lower frequency applications. An example application would be characterization of video waveforms in the presence of IF noise. Because most of the signal content is below the 8 MHz range, the 20 MHz noise filter is ideal. Many standard noise tests specify a 20 MHz noise bandwidth.

Antialias Filter

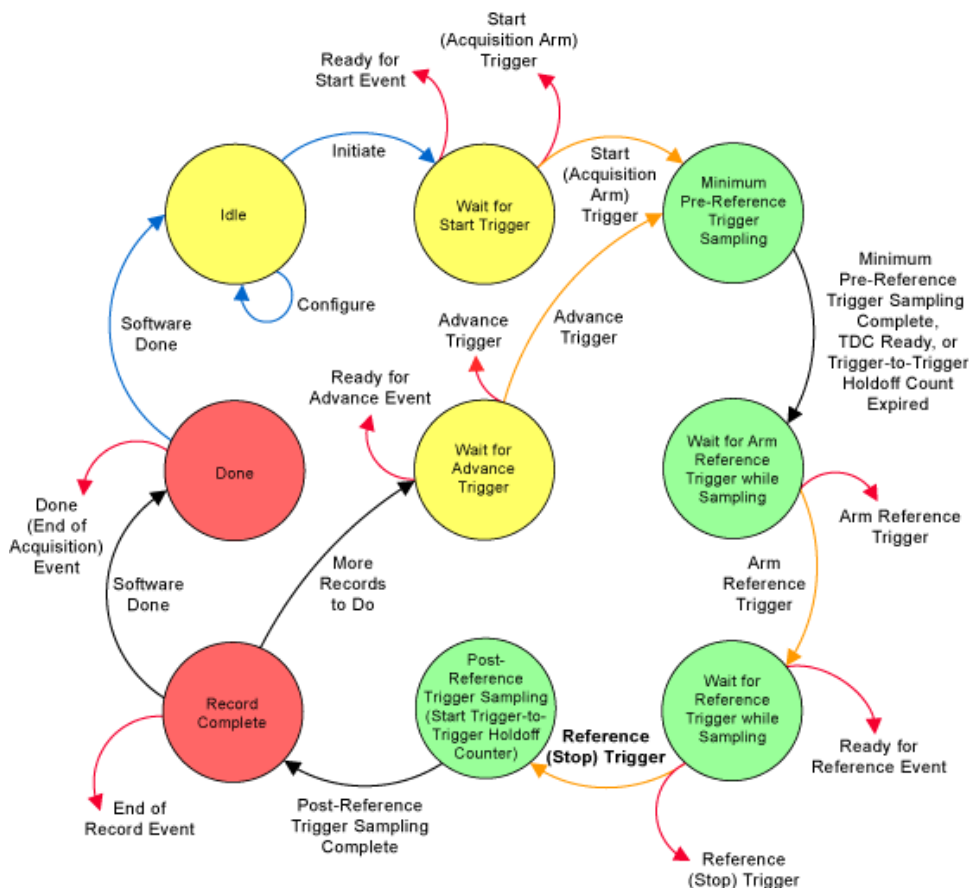
The NI 5122/5142 provides another bandwidth-limiting filter called an antialias filter. Although this filter can be used to limit out of band noise in many applications, its primary purpose is to minimize [aliasing](#) effects caused by signals greater than [Nyquist](#) applied to the 100 MS/s ADC. Refer to the [hardware specifications document](#) for more information on this filter.

A typical response with the antialias filter is shown in the following two figures.



SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

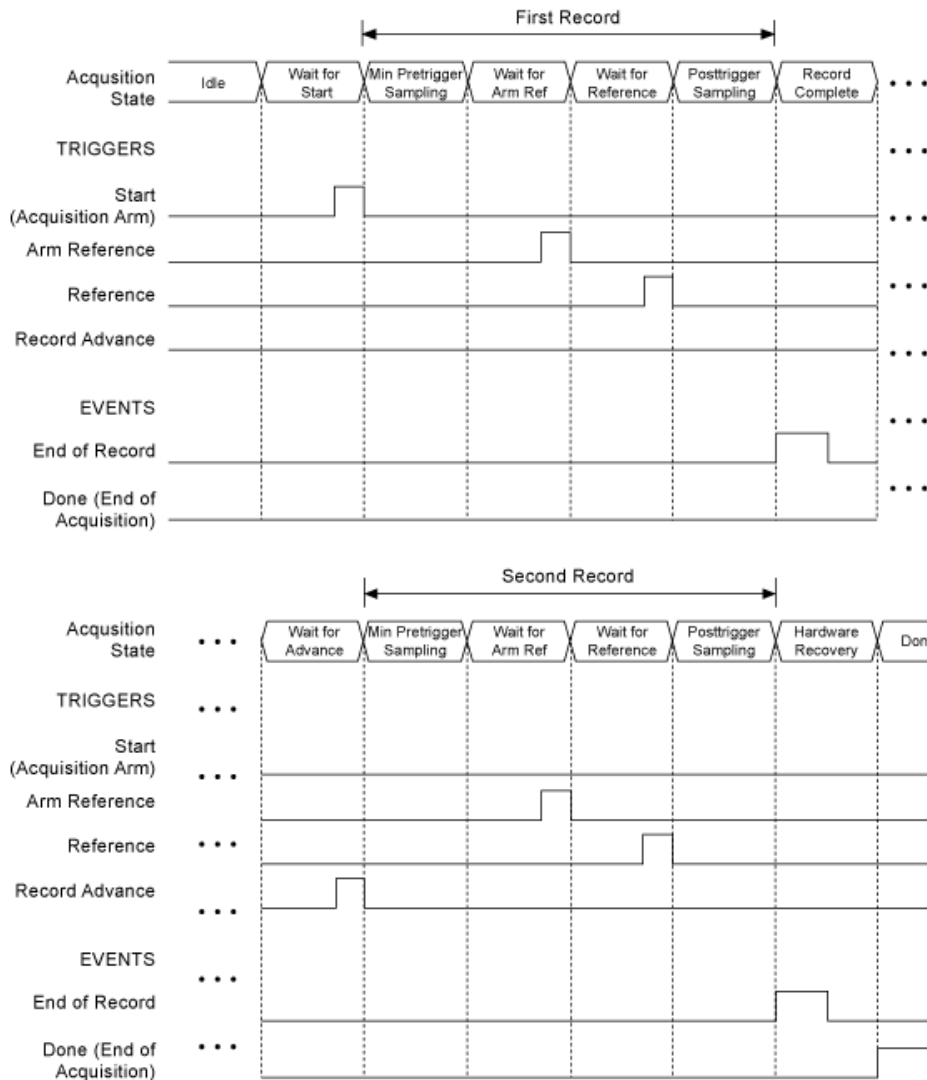
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5122/5124/5142 and NI PXIe-5122 Routing Matrix

The following table shows the signals available for export from the NI PXI-5122/5124/5142 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)
Exported Clocks			
Reference Clock (External)	√	√	√
Sample Clock (Full Rate)	—	√	—
Sample Clock (Decimated Rate)	√	√	√
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

NI PCI-5122/5124/5142 Routing Matrix

The following table shows the signals available for export from the NI PCI-5122/5124/5142, and the lines to which they can be routed.

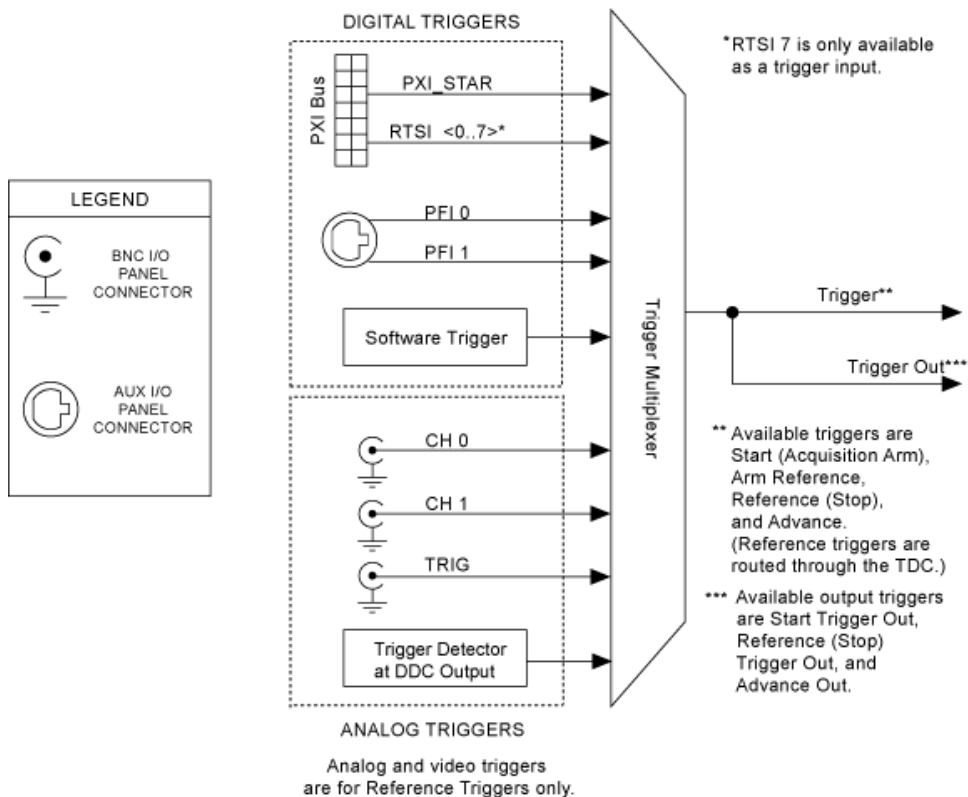
Source	Destination			
	RTSI <0..6> (RTSI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)	RTSI 7
Exported Clocks				
Reference Clock (External)	—	√	√	—
Reference Clock (Internal)	—	√	√	√
Sample Clock (Full Rate)	—	√	—	—
Sample Clock (Decimated Rate)	√	√	√	—
Triggers				
Acquisition Arm (Start) Trigger	√	—	√	—
Reference (Stop) Trigger	√	—	√	—
Events				
End of Record Event	√	—	√	—
End of Acquisition Event	√	—	√	—
Ready for Start Event	√	—	√	—
Ready for Reference Event	√	—	√	—
Ready for Advance Event	√	—	√	—

NI 5114/5122/5124/5142 Analog Trigger Types

The NI 5114/5122/5124/5142 supports the following analog trigger types: [edge](#), [hysteresis](#), [window](#), and [video](#) triggers.

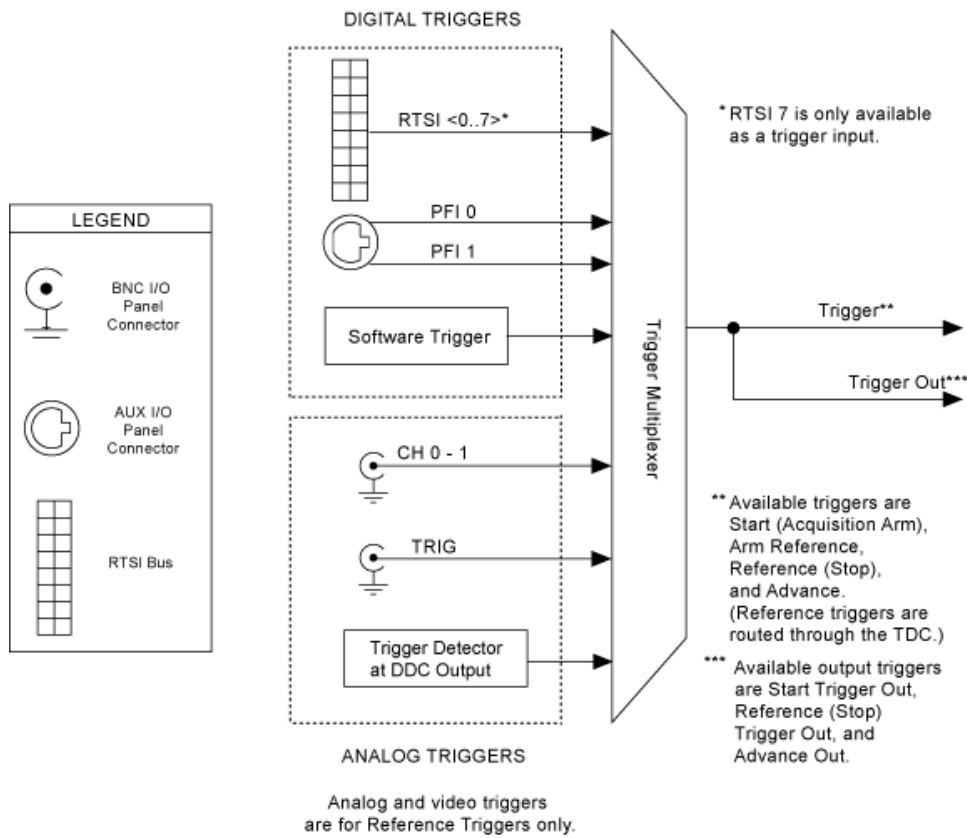
NI PXI-5142 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5142.



NI PCI-5142 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5142.



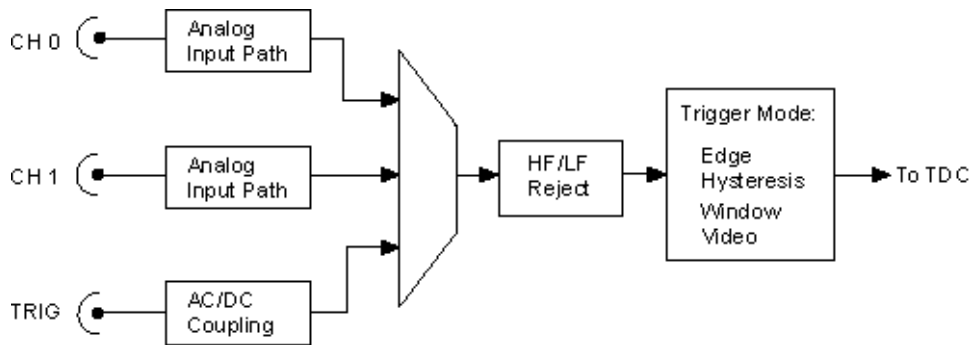
NI 5114/5122/5124/5142 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5114/5122/5124/5142 provides flexible, high-precision, low-jitter triggering features.

The NI 5114/5122/5124/5142 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5114/5122/5124/5142.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including

overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI 5122/5124/5142/5622/5922 Trigger Delay

[Trigger delay](#), which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

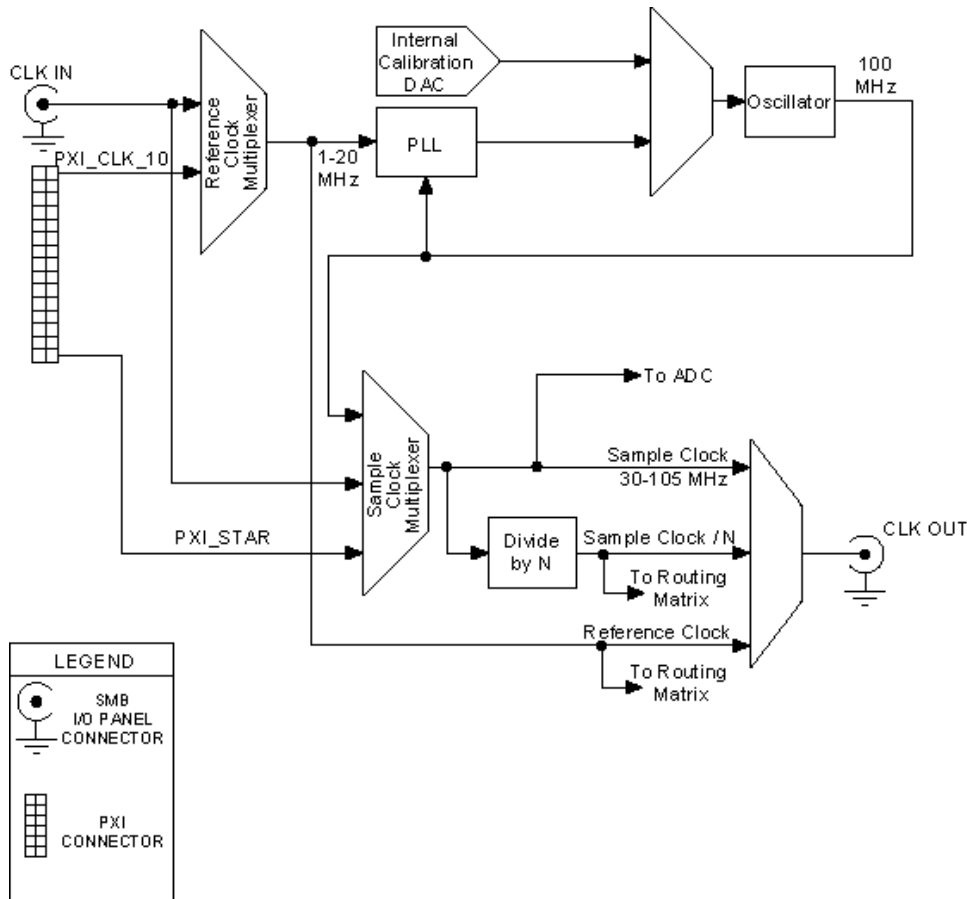
Max trigger delay in seconds = $[(2^{34} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock. The maximum trigger delay also changes based on the binary sample width. When binary sample width is set to 8, the maximum trigger delay is doubled. When binary sample width is set to 32, the maximum trigger delay is halved.

NI PXI-5122/5142 and NI PXIe-5122 Clocking

The clock circuitry on the NI PXI-5122/5142 and the NI PXIe-5122 offer versatile clocking options with the ability to use either the internal 100 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on these digitizers to phase lock the internal 100 MHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of these digitizers.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The digitizers can decimate their sample clocks (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

These digitizers have an onboard voltage controlled crystal oscillator (VCXO) running at 100 MHz. When using the onboard 100 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 100 MHz frequency of the VCXO. In PLL mode, the device phase locks its 100 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 100 MHz clock. In these cases these digitizers can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector or by routing the signal on the PXI backplane over the [PXI star](#) trigger line. Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The [reference clock](#) is used in the digitizer's phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizers can accept a reference clock from the front panel (CLK IN) as well as from PXI_CLK10. This reference clock can be any frequency from 5 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 100 MS/s when using the internal sample clock, the digitizer decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PXI-5122/5142 can only store one out of every two samples received from the ADCs to onboard memory. They can sample at rates of $100/n$ MS/s, where n is an integer value between 1 and 2^{16} . Because the digitizer does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When sampling at the

maximum rate, a free-running version of the actual sample clock is exported to the front panel of the digitizer. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.



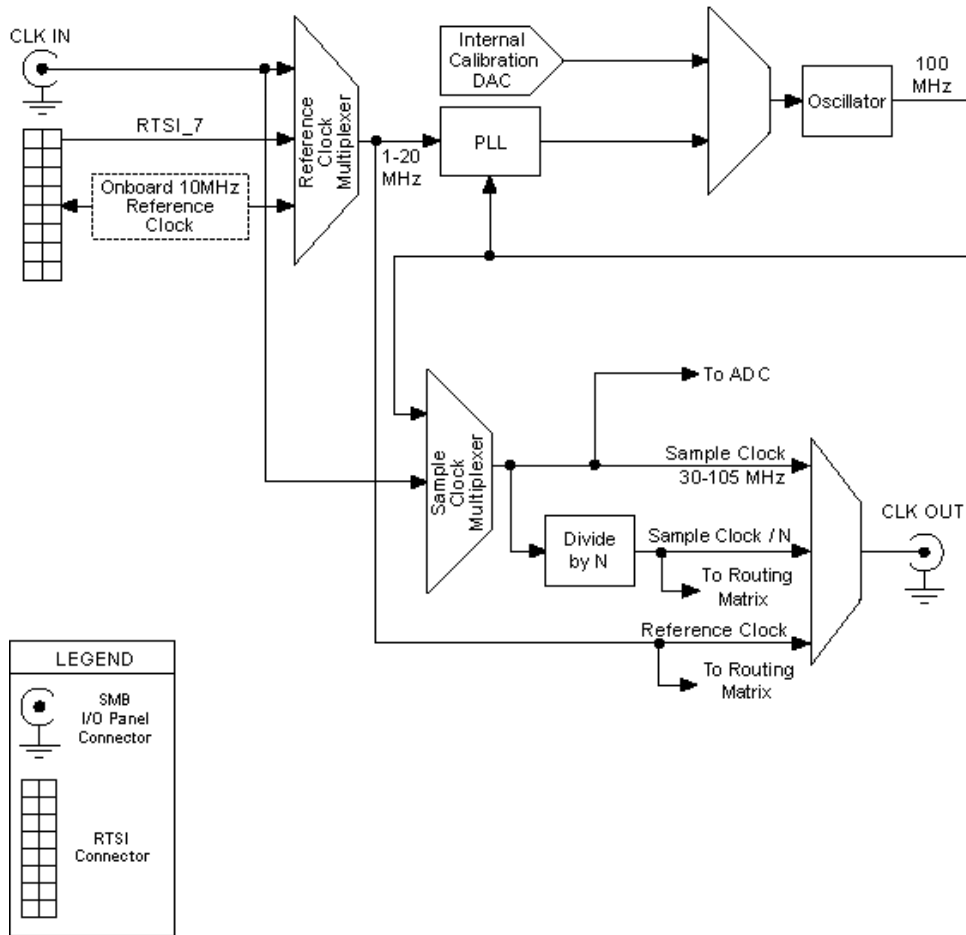
Note The NI PXI-5142 cannot export its sample clock while operating in [DDC](#) mode. If you want to export the sample clock, set the [DDC Enabled](#) property to False (default).

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXI-5122/5124/5142 and NI PXIe-5122 Routing Matrix](#).

NI PCI-5122/5142 Clocking

The clock circuitry on the NI PCI-5122/5142 offers versatile clocking options with its ability to use either the internal 100 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the NI PCI-5122/5142 to phase lock the internal 100 MHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5122/5142.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI PCI-5122/5142 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5122/5142 has an onboard voltage controlled crystal oscillator (VCXO) running at 100 MHz. When using the onboard 100 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 100 MHz frequency of the VCXO. In PLL mode, the NI PCI-5122/5142 phase locks its 100 MHz sample clock to the supplied reference clock. The PLL mode is

useful when synchronizing the NI PCI-5122/5142 with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 100 MHz clock. In these cases the NI PCI-5122/5142 can accept an external sample clock. External clocking also provides a method to synchronize the NI PCI-5122/5142 to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the NI PCI-5122/5142 from the front panel connector. Refer to the specifications document for your device for external sample clock requirements.



Note When an external sample clock is used, the [TDC](#) is disabled. Refer to the device specifications for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The reference clock is used in the NI PCI-5122/5142 PLL circuit to synchronize the sample clock to the reference clock. The NI PCI-5122/5142 can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 5 MHz to 20 MHz if it is provided to CLK IN or to RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PCI-5122/5142 reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 100 MS/s when using the internal sample clock, the NI PCI-5122/5142 decimates the sampled data. When you set a sampling rate of 50 MS/s, the NI PCI-5122/5142 only stores one out of every two samples it receives from the ADCs to onboard memory. The NI PCI-5122/5142 can sample at rates of $100/n$ MS/s, where n is an integer value between 1 and (2^{16}) . Because the

NI PCI-5122/5142 does not change the frequency of the clock sent to the ADCs, it creates a sample clock for export based on the effective sampling rate that you have configured. When sampling at the maximum rate, a free-running version of the actual sample clock is exported to the front panel of the NI PCI-5122/5142. When the data is being decimated, a divided down version of the sample clock is exported. The divided down version of the sample clock only toggles while data is being acquired, and is driven low after the acquisition completes.

Exporting Reference Clock

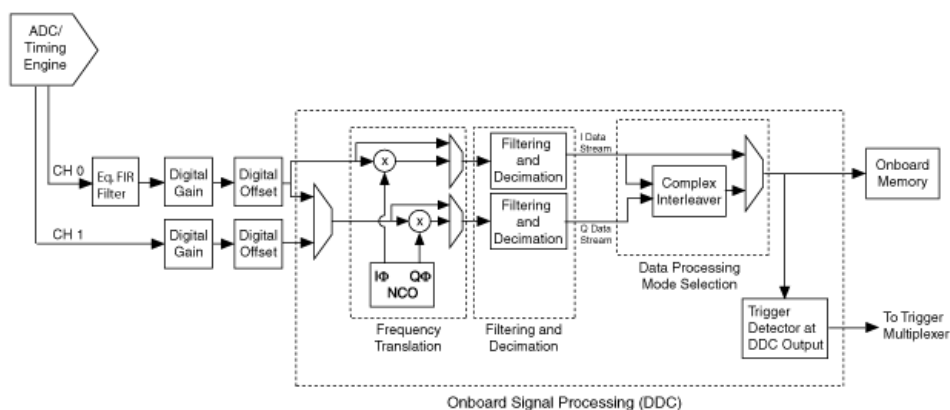
If you are using the internal reference clock you can export that same reference clock to other NI PCI-5122/5142 devices for synchronization purposes. For more information, refer to the section on synchronization. If you are using a reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5122/5124/5142 Routing Matrix](#).

NI 5142 Onboard Signal Processing (OSP)

The onboard signal processing (OSP) block is a general-purpose block of digital signal processing components.

The following figure shows the main components found in the OSP block of the NI 5142.

Click each area for more information about the main components of the OSP block.



Related Topics

[NI PXI-5142 Trigger Sources](#)

[NI PCI-5142 Trigger Sources](#)

NI 5142 Onboard Signal Processing Components

- [Equalization FIR Filter](#)
- [Digital Gain](#)
- [Digital Offset](#)
- [Digital Downconverter \(DDC\)](#)
- [Frequency Translation](#)
- [Filtering and Decimation](#)
- [Data Processing Mode Selection](#)
- [Onboard Signal Processing Triggering](#)

Digital Gain

Digital gain can be used to change the gain of each channel. You can program the digital gain of each channel independently using the [Digital Gain](#) attribute. The gain can range from -1.5 to $+1.5$ (unitless). You can use the digital gain to attenuate the channel data to eliminate overflows in the OSP block. To eliminate overflows, the digital gain should attenuate the ADC data (it should be set to a value less than 1). The digital gain circuit itself can overflow if the following condition is not met:

$$-1 \leq \text{Unity Scaled User Data} \times \text{Pre-Filter Gain} \leq 1$$

If an overflow occurs, the data is clipped and NI-SCOPE returns an error. To prevent data clipping, attenuate the waveform data or reduce the digital gain.



Tip To change the default error reporting caused by OSP overflows, change the [Overflow Error Reporting](#) attribute.

You can also use Digital Gain to do fine-gain adjustments on the ADC data to compensate for channel or system gain imperfections. These adjustments occur in

hardware before the data is stored to memory, which saves an additional processing step that would normally occur in software.

Digital Offset

The digital offset can be used to change the offset of each channel. You can program the digital offset of each channel independently using the [Digital Offset](#) attribute. The offset can range from $-(\mathbf{Vertical\ Range} \times 0.4)$ to $+(\mathbf{Vertical\ Range} \times 0.4)$. The digital offset circuit can overflow if the following condition is not met:

$$-(\mathbf{Vertical\ Range}/2) \leq \mathbf{Channel\ Data} + \mathbf{Digital\ Offset} \leq +\mathbf{Vertical\ Range}/2.$$

If an overflow occurs, the data is clipped and NI-SCOPE returns an error. To prevent data clipping, attenuate the waveform data or reduce the digital offset.



Tip To change the default error reporting caused by OSP overflows, change the [Overflow Error Reporting](#) attribute.

Digital Downconverter (DDC)

The DDC takes a programmable portion of the [Nyquist](#) spectrum of the ADC and moves it down to baseband (centered around 0 Hz). This process is performed by first [frequency translating](#) the center of the band of interest to 0 Hz. The data is then [filtered and decimated](#) to get alias-protected data of the requested bandwidth. The data output from the DDC is stored into memory at this lower sample rate as complex IQ pairs.

Equalization FIR Filter

Equalization FIR Filter is a generic FIR filter. You can download coefficients to the filter using the [niscope Configure Equalization Filter Coefficients VI](#) or the [niScope_ConfigureEqualizationFilterCoefficients](#) function.

The number of coefficients you can download to the filter is determined by the [Equalization Num Coefficients](#) property or the [NISCOPE_ATTR_EQUALIZATION_NUM_COEFFICIENTS](#) attribute. After the coefficients are downloaded, you can use the [Equalization Filter Enabled](#) property or the [NISCOPE_ATTR_EQUALIZATION](#)

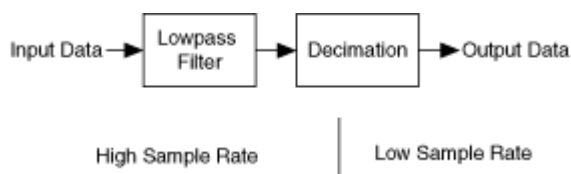
`FILTER_ENABLED` attribute to enable the filter in the hardware. The Equalization FIR Filter can overflow if the data values exceed the vertical range.

Frequency Translation

Frequency translation moves the [Nyquist](#) frequency spectrum of the acquired signal by the amount specified by the [Center Frequency](#) attribute. This is accomplished by multiplying the acquired data stream with the output of the numerically controlled oscillator (NCO). The NCO is a digital circuit that creates two sine waves of the same frequency (the Center Frequency) with two independent phases. You can use the Center Frequency attribute to set the carrier frequency. The I phase cosine waveform is multiplied by the I signal path, and the Q phase cosine waveform is multiplied by the Q data path. The I and Q phases are programmable from 0° to 360° by setting the [Frequency Translation Phase I](#) and [Frequency Translation Phase Q](#) attributes/properties. Both the frequency and the phases can be updated during acquisition. You can consider the values of the Frequency Translation Phase I/Q attributes as the start phase of the NCO waveforms. The NCO outputs start at the specified phase values when the start trigger is received. Normally, the I and Q phases would be set to 0° and 90° , respectively, but the flexibility allows you to correct any externally-induced impairments in the IF signal.

Filtering and Decimation

The filtering and decimation stage of the OSP block reduces the effective sample rate of the digitizer while protecting the frequency spectrum of the decimated data from aliases. This protection occurs when the data passes through a lowpass filter before decimation. The normal decimation in digitizers (when not using OSP) does not protect the frequency spectrum of the decimated data from aliases.



Data Processing Mode Selection

The Data Processing Mode Selection stage selects between Real and Complex processing. In Real mode, only the Real I data is returned when a waveform fetch is executed. In Complex mode, the I and Q data streams are interleaved before they are stored to memory. When the complex data is fetched, the data is returned as interleaved I and Q data. The data can be fetched as an array of complex IQ samples, as an array of interleaved real I and Q samples, or as separate arrays of real I and Q samples.

Onboard Signal Processing Triggering

In [DDC](#) mode, the NI 5142 has the ability to analog trigger from circuitry located on the daughtercard, or from the OSP circuit located at the output of the DDC. The analog circuit on the daughtercard always triggers on data before any onboard signal processing occurs. The OSP circuit always triggers on data after onboard signal processing occurs.

In [DDC](#) mode, the NI 5622 has the ability to analog trigger only from the OSP circuit located at the output of the DDC. The OSP circuit always triggers on data after onboard signal processing occurs.

When the DDC is in [complex mode](#), the OSP circuit triggers on the magnitude of both the I stream and the Q stream. The circuit uses the following formula:

$$\text{Sqrt}(I^2 + Q^2)$$

This magnitude is proportional to the power of the signal. By default, the device uses the analog circuit on the daughtercard. You can change the trigger used by the device by using the [Ref Trigger Detector Location](#) property or the `NISCOPE_ATTR_REF_TRIGGER_DETECTOR_LOCATION` attribute.

The OSP trigger circuit can also trigger on the beginning of [bursts of data](#). To enable this functionality, configure the [Ref Trigger Minimum Quiet Time](#) property or the `NISCOPE_ATTR_REF_TRIGGER_MINIMUM_QUIET_TIME` attribute to the value of time that corresponds to the amount of time that the signal must be below the trigger level to constitute a quiet time. For more information, refer to [Burst Triggers](#).

OSP Basic Properties/Attributes

The following topics contain overviews of the basic OSP properties/attributes.

- [Center Frequency](#)
- [Data Processing Mode](#)
- [DDC Enabled](#)
- [Frequency Translation Enabled](#)
- [Q Source](#)
- [Frequency Translation Phase I](#)
- [Frequency Translation Phase Q](#)
- [Overflow Error Reporting](#)
- [Digital Gain](#)
- [Digital Offset](#)

DDC Enabled

The DDC Enabled property/attribute activates the functionality of the DDC block. To use any of the features in the DDC block, you must set DDC Enabled to TRUE.

LabVIEW	C/C++
DDC Enabled	<code>NISCOPE_ATTR_DDC_ENABLED</code>



Tip To achieve maximum flatness in the filter response of the device in DDC mode, set the maximum input frequency to full bandwidth (-1). However, keep in mind that this action does not protect against aliasing frequency content of the input signal above ($0.5 \times$ Sample Clock Timebase Rate). When using internal clocking, the Sample Clock Timebase Rate is 100 MS/s (NI 5142) or 150 MS/s (NI 5622).

Hardware Calibration of Binary Data

In a normal acquisition, you cannot assume that the binary data maps perfectly to the vertical range. For example, if you fetch 16-bit binary data, you might expect that the positive full scale binary value corresponds to the maximum positive voltage of the vertical range, and that the negative full scale binary value corresponds to the maximum negative voltage. This is not the case, however, for a

number of reasons. First, the full scale of the ADC is not mapped to the vertical range. A few codes on the positive and negative ends are left as headroom, in case the input signal slightly exceeds the specified vertical range. Second, the binary values do not account for the required corrections to gain and offset based on calibration data. If you fetch binary data, you can convert to voltage values using the gain and offset values from the `wfmInfo` struct. These values account for the ADC normalization and calibration data.

$$\text{Voltage} = (\text{Binary Value} \times \text{wfmInfo.gain}) + \text{wfmInfo.offset}$$

When the DDC is enabled, the onboard signal processing block accounts for both ADC normalization and calibration data. So the waveform samples are stored and can be fetched as calibrated, normalized binary data. This means that the range of binary values maps exactly to the vertical range you configure.

$$\text{Voltage} = \text{Binary Value} \times \text{Range} / 2^{(\text{ADC resolution in bits})}$$

The gain and offset values in the `wfmInfo` struct work just as well in this case (but they do not hold normalization and calibration information), so the following conversion still works:

$$\text{Voltage} = (\text{Binary Value} \times \text{wfmInfo.gain}) + \text{wfmInfo.offset}$$

In a normal acquisition, you must know the gain and offset values to correctly convert the binary data to voltage values. In DDC mode, you can assume the following:

- The maximum positive binary value maps to the maximum positive voltage of the vertical range.
- The maximum negative binary value maps to the maximum negative voltage of the vertical range.
- The vertical range is divided evenly across the possible binary values.

These assumptions may be useful for streaming or logging binary data, because you do not need to know the gain and offset value to recover the measured voltages.

Center Frequency

The Center Frequency property/attribute indicates the frequency by which the frequency translation stage of the DDC block frequency translates the input data; that is, the center frequency of the region of the spectrum that you want to downconvert to baseband.

LabVIEW Property	C/C++ Attribute
<u>Center Frequency</u>	<u>NISCOPE_ATTR_CENTER_FREQUENCY</u>

Frequency Translation Enabled

The Frequency Translation Enabled property enables frequency translating the data around the user-selected center frequency down to baseband.

LabVIEW Property	C/C++ Attribute
<u>Frequency Translation Enabled</u>	<u>NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_ENABLED</u>

Data Processing Mode

The Data Processing Mode property/attribute determines how the DDC block processes data. If Data Processing Mode is set to Real, only the I data stream is enabled and the waveform data points are real numbers (I data). If Data Processing Mode is set to Complex, both the I and Q data streams are enabled and interleaved, and waveform data points are complex numbers (IQ data).

LabVIEW Property	C/C++ Attribute
<u>Data Processing Mode</u>	<u>NISCOPE_ATTR_DATA_PROCESSING_MODE</u>

Frequency Translation Phase I

The Frequency Translation Phase I property/attribute specifies the I oscillator phase of the sine wave in degrees at the first point acquired.

LabVIEW Property	C/C++ Attribute
------------------	-----------------

Frequency Translation Phase INISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHASE_I

Frequency Translation Phase Q

The Frequency Translation Phase Q property/attribute specifies the Q oscillator phase of the sine wave in degrees at the first point acquired. Use this property only when the Data Processing Mode is set to Complex.

LabVIEW PropertyC/C++ AttributeFrequency Translation Phase QNISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHASE_Q

Overflow Error Reporting

The Overflow Error Reporting property/attribute configures error reporting when the DDC block detects an overflow in any of its stages. All overflows lead to clipping of the waveform.

LabVIEW PropertyC/C++ AttributeOverflow Error ReportingNISCOPE_ATTR_OVERFLOW_ERROR_REPORTING

Q Source

The Q Source property/attribute indicates the channel that is the input of the Q data stream of the DDC.

LabVIEW PropertyC/C++ AttributeQ SourceNISCOPE_ATTR_DDC_Q_SOURCE

Digital Gain

The Digital Gain property/attribute applies gain to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset

LabVIEW Property	C/C++ Attribute
Digital Gain	<code>NISCOPE_ATTR_DIGITAL_GAIN</code>

Digital Offset

The Digital Offset property/attribute applies offset to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset

LabVIEW Property	C/C++ Attribute
Digital Offset	<code>NISCOPE_ATTR_DIGITAL_OFFSET</code>

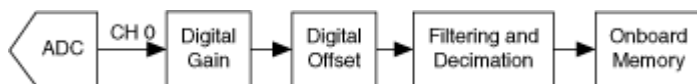
Common DDC Applications (NI 5142)

The DDC block of the NI 5142 is useful for the following applications:

- [Alias-Protected Decimation](#)
- [AM Demodulation](#)
- [Quadrature Downconversion](#)
- [Baseband Decimation](#)

To use the NI 5142 in normal digitizer mode, set the [DDC Enabled](#) property to False (default).

Alias-Protected Decimation

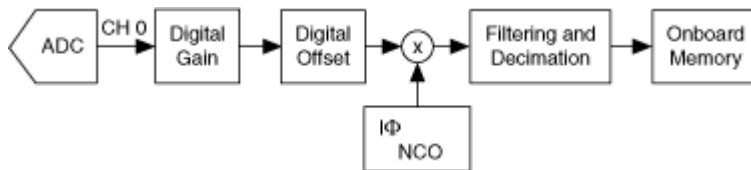


For alias-protected decimation applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Real.

3. Set the [Min Sample Rate](#) property to the desired decimated sample rate.
4. Set the [Frequency Translation Enabled](#) property to False.

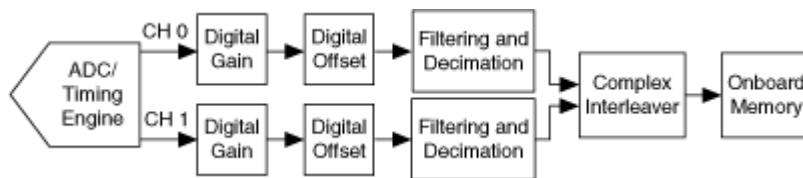
AM Demodulation



For AM demodulation applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Real.
3. Set [Min Sample Rate](#) property to the desired baseband sample rate.
4. Set [Frequency Translation Enabled](#) property to True.
5. Set [Center Frequency](#) property to the carrier frequency of the AM signal you want to downconvert.

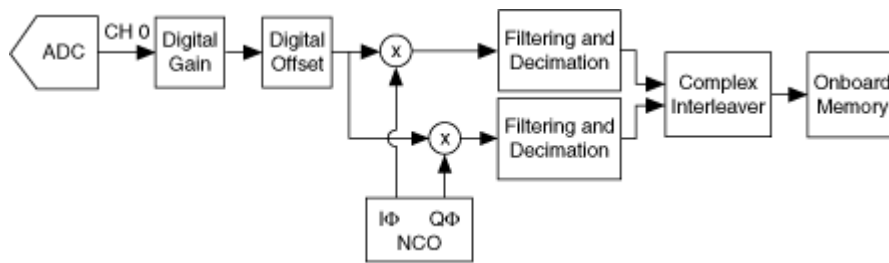
Baseband Decimation



For baseband decimation applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Complex.
3. Set the [Min Sample Rate](#) property to the desired baseband sample rate.
4. Set the [Frequency Translation Enabled](#) property to False.
5. Set the [Q Source](#) property to "1".

Quadrature Downconversion



For quadrature downconversion applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Complex.
3. Set the [Min Sample Rate](#) property to the desired IQ data rate.
4. Set the [Frequency Translation Enabled](#) property to True.
5. Set the [Center Frequency](#) property to the desired center frequency of the IF band you want to downconvert.

NI 5122/5124/5142 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5122/5124/5142, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges, both input impedance paths (50 Ω and 1 MΩ), and all filter paths (enabled/disabled).
- Calibrates trigger level offset and gain.
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Compensates 1 MΩ input frequency flatness, then compensates input capacitance so that it is equal regardless of selected attenuator range.
- Takes approximately 2 minutes to complete.

Using the example as a comparison, the NI 5122/5124/5142, using self-calibration and a 5 V input, will have an error at 48 °C of

$$5 \text{ V}(0.65\% \text{ of input} + 8 \text{ mV}) = 40.5 \text{ mV error}$$

This result compares favorably to the 180 mV error for a typical digitizer as described in the example without self-calibration.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5122/5124/5142 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

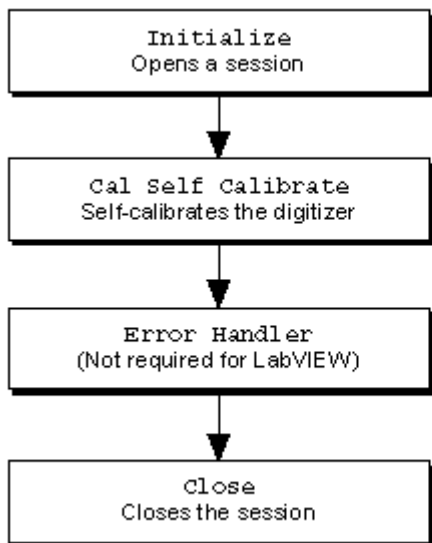
Input Connections During Self-Calibration

The NI 5122/5124/5142 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger level Trigger timing AC flatness Input capacitance	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy Optimizes performance with external 10:1 probes
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5122/5142 Onboard Memory

The NI 5122/5142 allocates at least 384 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is approximately 128 14-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 128 samples. Refer to the [hardware specifications document](#) for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5122/5142 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

NI 5152

Features

The NI 5152 has the following features:

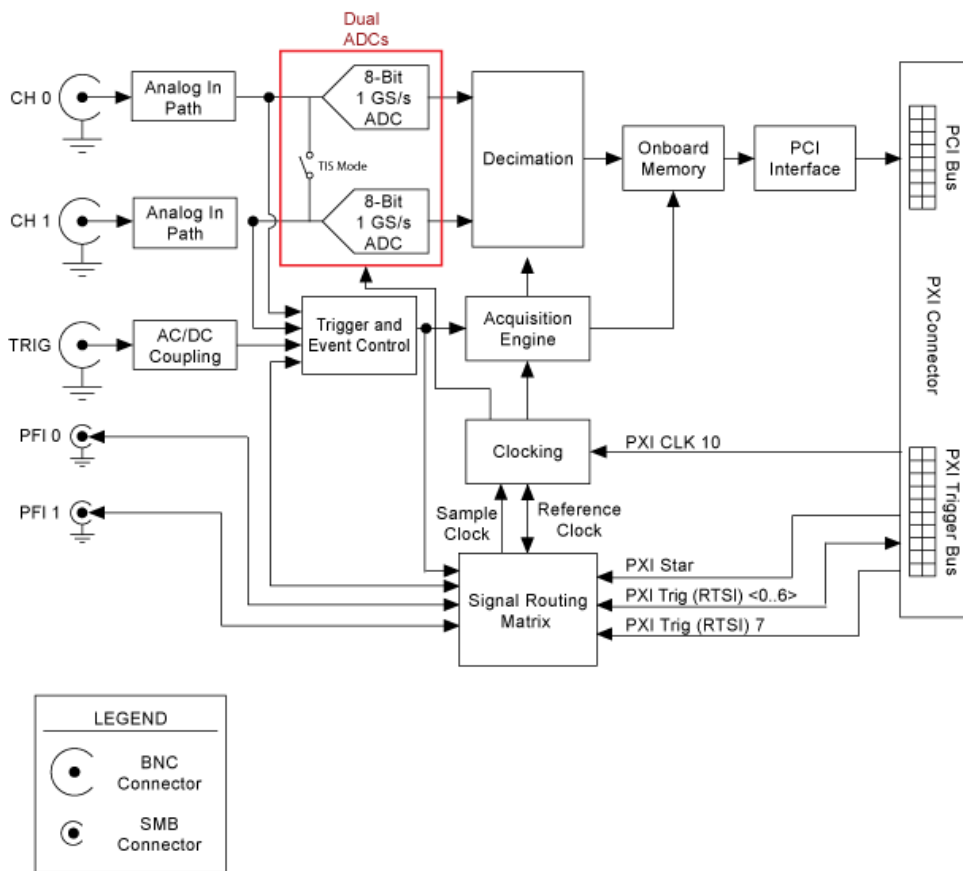
- 2 channels, simultaneously sampled
- 8-bit vertical resolution
- 1 GS/s real-time sampling rate, 2 GS/s time interleaved sampling rate
- 300 MHz bandwidth
- PXI or PCI versions
- 8 MB, 64 MB, or 256 MB of memory per channel
- [NI-TClk synchronization](#)

Related Topics

[Features Supported by Device](#)

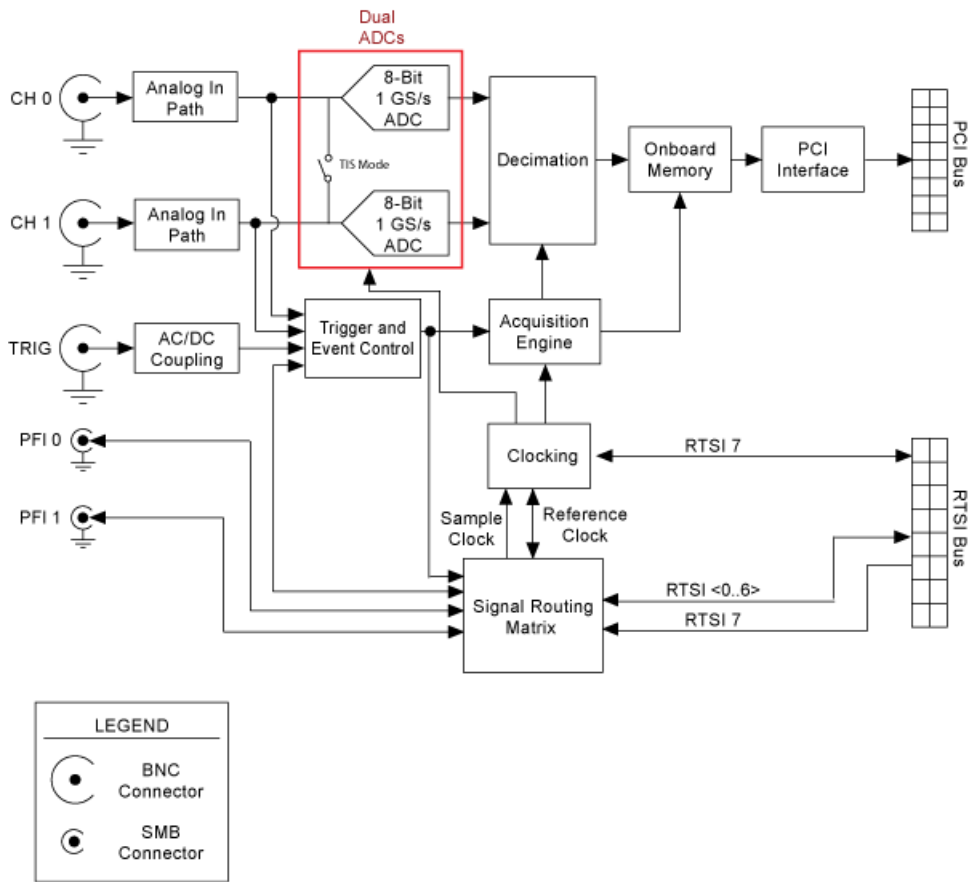
NI PXI-5152 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5152. The dual ADC architecture enables the [Time Interleaved Sampling \(TIS\)](#) feature of these digitizers.



NI PCI-5152 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5152. The dual ADC architecture enables the Time Interleaved Sampling (TIS) feature of these digitizers.



PXI-5152 Front Panel

The following figure shows the front panel of the PXI-5152. Descriptions of the LEDs and connectors are shown below.



LEDs

The PXI-5152 has two LEDs to indicate status: Access and Active.


Access LED

The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.
Green	The module is ready to be programmed by NI-SCOPE.

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ■ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ■ External sample clock error—The module is unable to detect the external sample clock. ■ Overvoltage error—The module has detected a 50 Ω overvoltage error. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>

Connectors

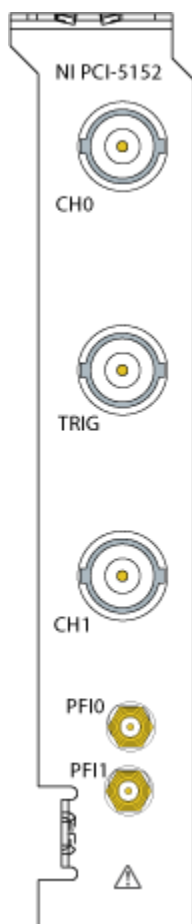
The PXI-5152 has the following connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions

TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
PFI 0	Standard SMB jack	Reference clock in, sample clock in, digital trigger in/out
PFI 1	Standard SMB jack	Reference clock out, +5 V out, digital trigger in/out

PCI-5152 Front Panel

The following figure shows the front panel of the PCI-5152. Descriptions of the connectors are shown below.



Connectors

The PCI-5152 has the following connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
PFI 0	Standard SMB jack	Reference clock in, sample clock in, digital trigger in/out
PFI 1	Standard SMB jack	Reference clock out, +5 V out, digital trigger in/out

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

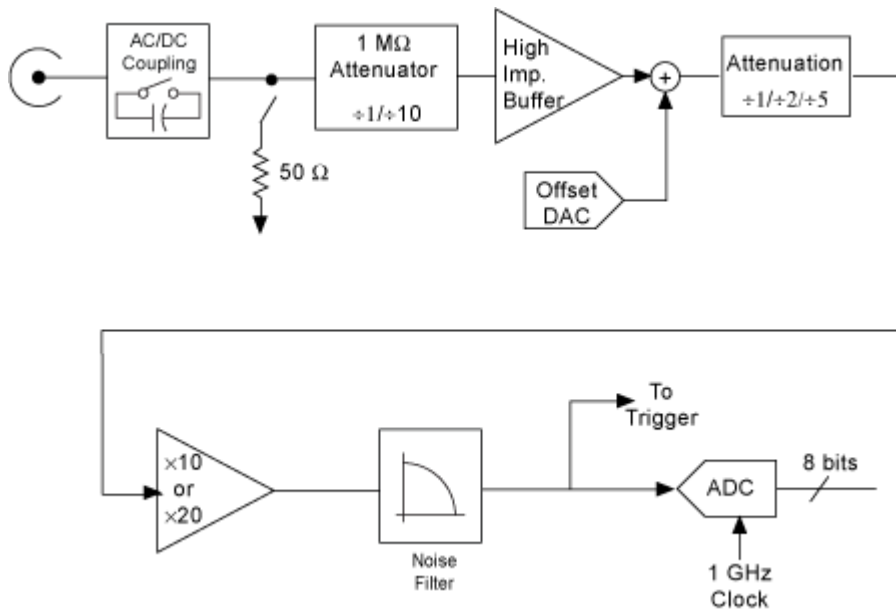
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5152 Input Signal Conditioning

The NI 5152 provides two independent digitizer input channel signal conditioning paths. Each path provides you with a choice of 50 Ω input impedance or 1 M Ω input impedance, as shown in the following diagram.



Note The ground on the device inputs is connected to the chassis ground.

NI 5152 Input Ranges

The NI 5152 supports the following input ranges: 0.1, 0.2, 0.4, 1, 2, and 10 V_{pk-pk}

NI 5152 Input Impedance

You can set the NI 5122/5124/5142/5152 analog input impedance to either 50 Ω or 1 MΩ. The 1 MΩ path is required in applications that require minimal loading or that require using a standard 10:1 oscilloscope probe.

Protection

The 50 Ω inputs of the NI 5152 are protected by a thermal disconnect circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω.

NI 5152 AC/DC/GND Coupling

You can select AC, DC, or GND [input coupling](#) for the 1 M Ω input path or the 50 Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5152 Vertical Offset

The following table shows the valid vertical offset for each range on the NI 5152.

Range	50 Ω Vertical Offset	1 M Ω Vertical Offset
0.1 V _{pk-pk}	± 1 V	± 1 V
0.2 V _{pk-pk}	± 1 V	± 1 V
0.4 V _{pk-pk}	± 1 V	± 1 V
1 V _{pk-pk}	± 1 V	± 1 V
2 V _{pk-pk}	± 6 V	± 10 V
4 V _{pk-pk}	± 5 V	± 10 V
10 V _{pk-pk}	± 2 V	± 10 V

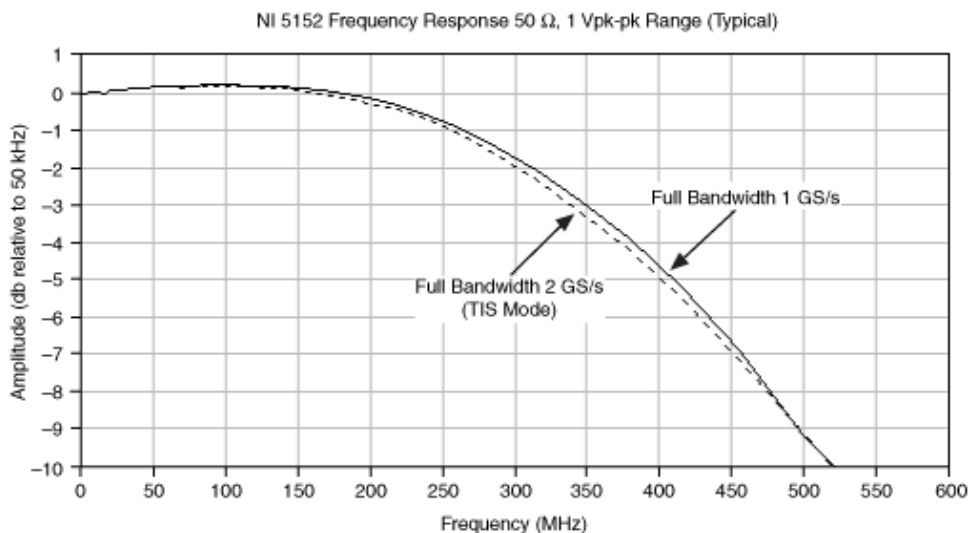
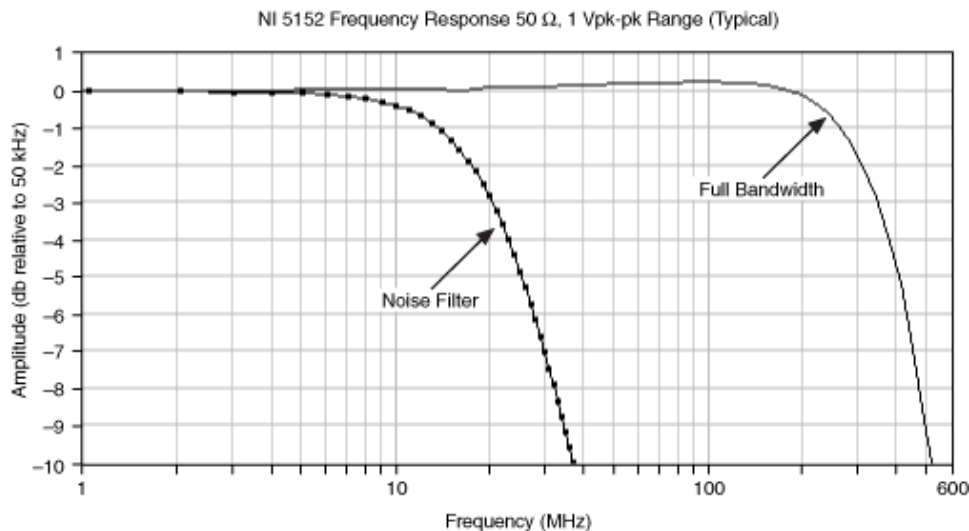


Note The maximum allowable DC voltage is 7 V in the 50 Ω path. The vertical offsets were chosen to stay within these voltage limits.

NI 5152 Noise Filter

The NI 5152 provides a 20 MHz noise filter that limits the bandwidth of the signal path through both the 1 M Ω and 50 Ω signal paths. This filter is intended to reduce noise when the signal content is 20 MHz or less.

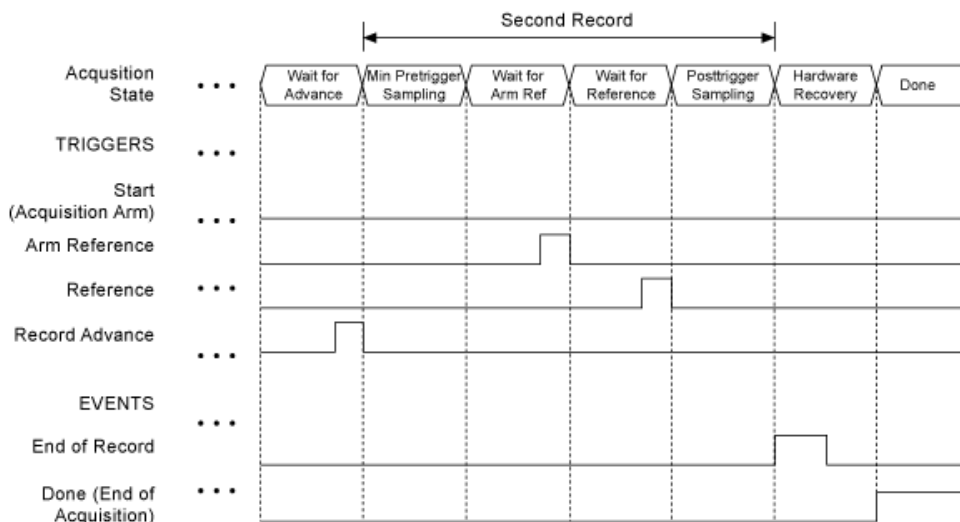
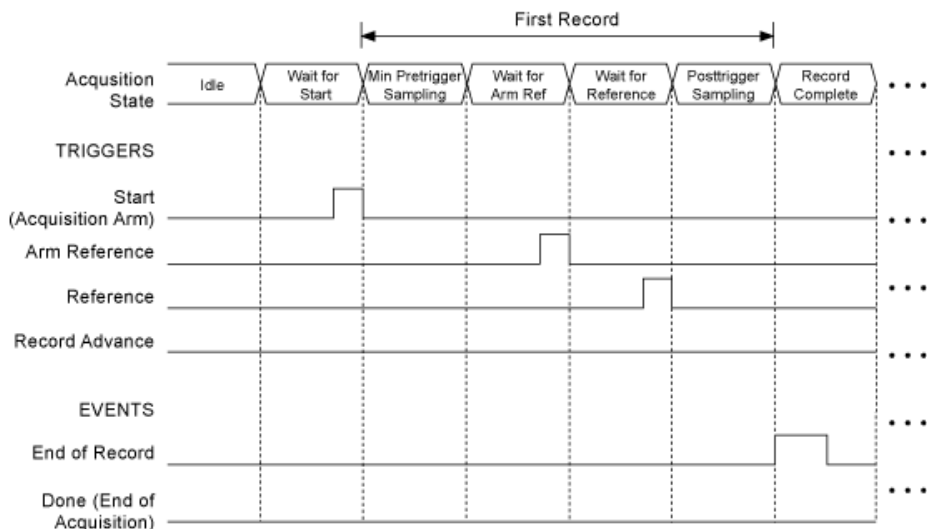
Typical frequency responses of the NI 5152 are shown in the following figures.



SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5152/5153/5154 Routing Matrix

The following table shows the signals available for export from the NI PXI-5152/5153/5154 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	PFI 0	PFI 1
Exported Clocks			
Reference Clock (External)	√	—	√
Triggers			
Acquisition Arm (Start) Trigger	√	√	√
Reference (Stop) Trigger	√	√	√
Events			
End of Record Event	√	√	√
End of Acquisition Event	√	√	√
Ready for Start Event	√	√	√
Ready for Reference Event	√	√	√
Ready for Advance Event	√	√	√

NI PCI-5152/5153/5154 Routing Matrix

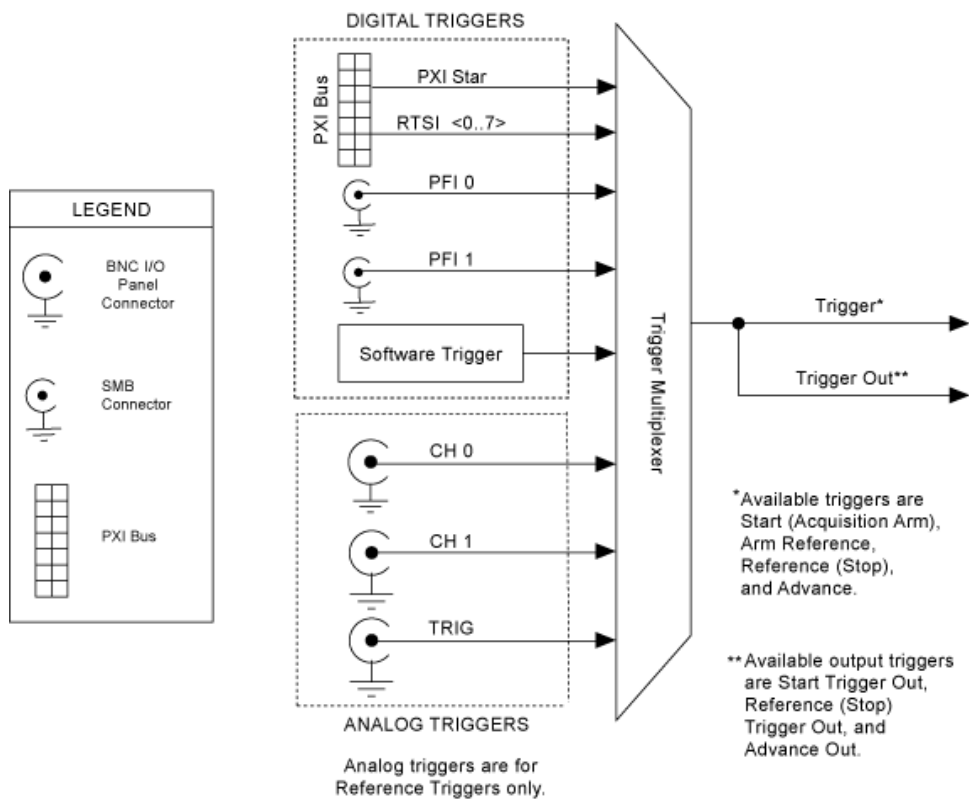
The following table shows the signals available for export from the NI PCI-5152/5153/5154 and the lines to which they can be routed.

Source	Destination			
	RTSI <0..6> (RTSI Bus)	PFI 0	PFI 1	RTSI 7

Exported Clocks				
Reference Clock (External)	√	—	√	—
Reference Clock (Internal)	—	√	√	√
Triggers				
Acquisition Arm (Start) Trigger	√	√	√	—
Reference (Stop) Trigger	√	√	√	—
Events				
End of Record Event	√	√	√	—
End of Acquisition Event	√	√	√	—
Ready for Start Event	√	√	√	—
Ready for Reference Event	√	√	√	—
Ready for Advance Event	√	√	√	—

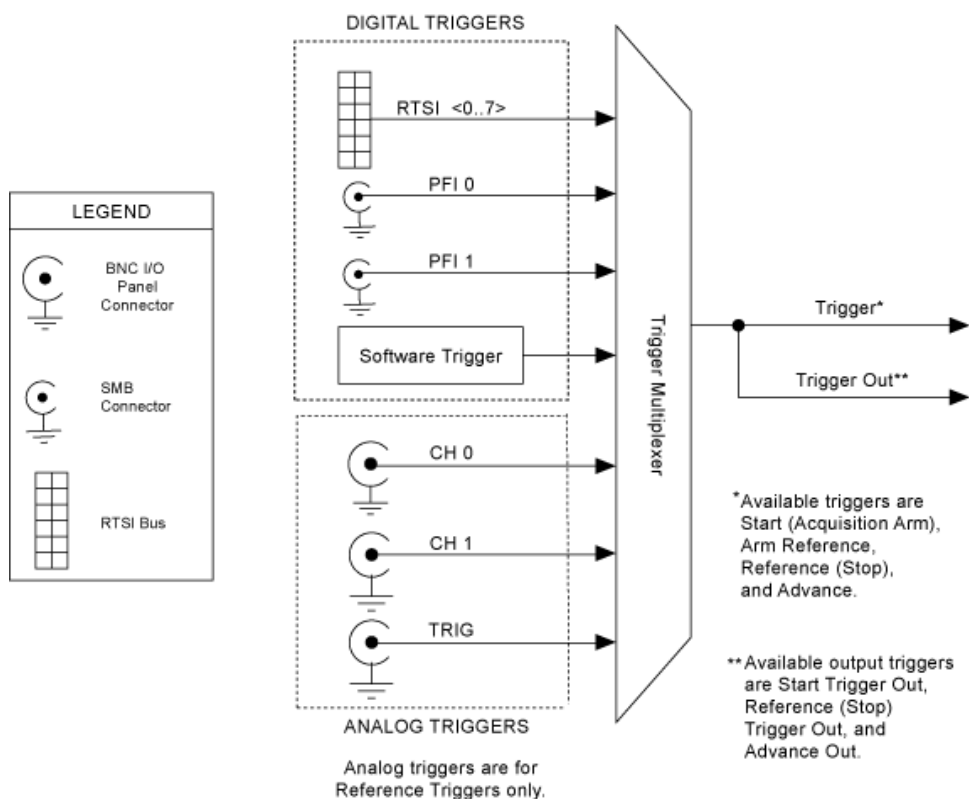
NI PXI-5152/5153/5154 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5152/5153/5154.



NI PCI-5152/5153/5154 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5152/5153/5154.



NI 5114/5152/5153/5154 Trigger Delay

Trigger delay, which is specified in seconds, is achieved on the NI 5114/5152/5153/5154 by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

$$\text{Max trigger delay in seconds} = [(2^{35} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$

Note The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock.

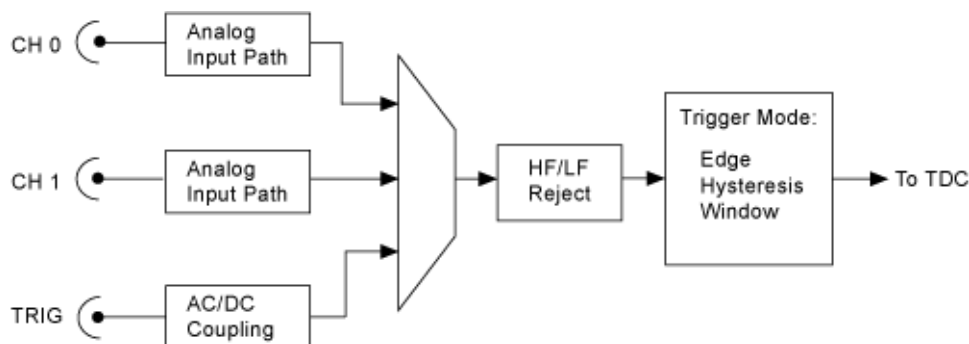
NI 5152/5153/5154 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5152/5153/5154 provides flexible, high-precision, low-jitter triggering features.

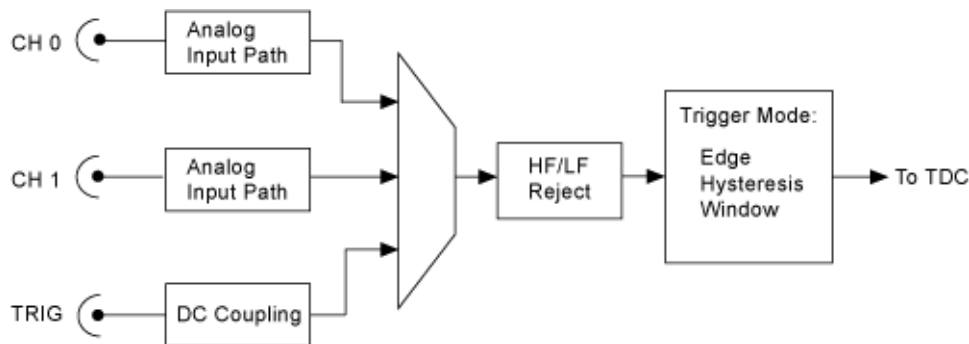
The NI 5152/5153/5154 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5152.



The following figure shows the analog trigger paths for the NI 5153/5154.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

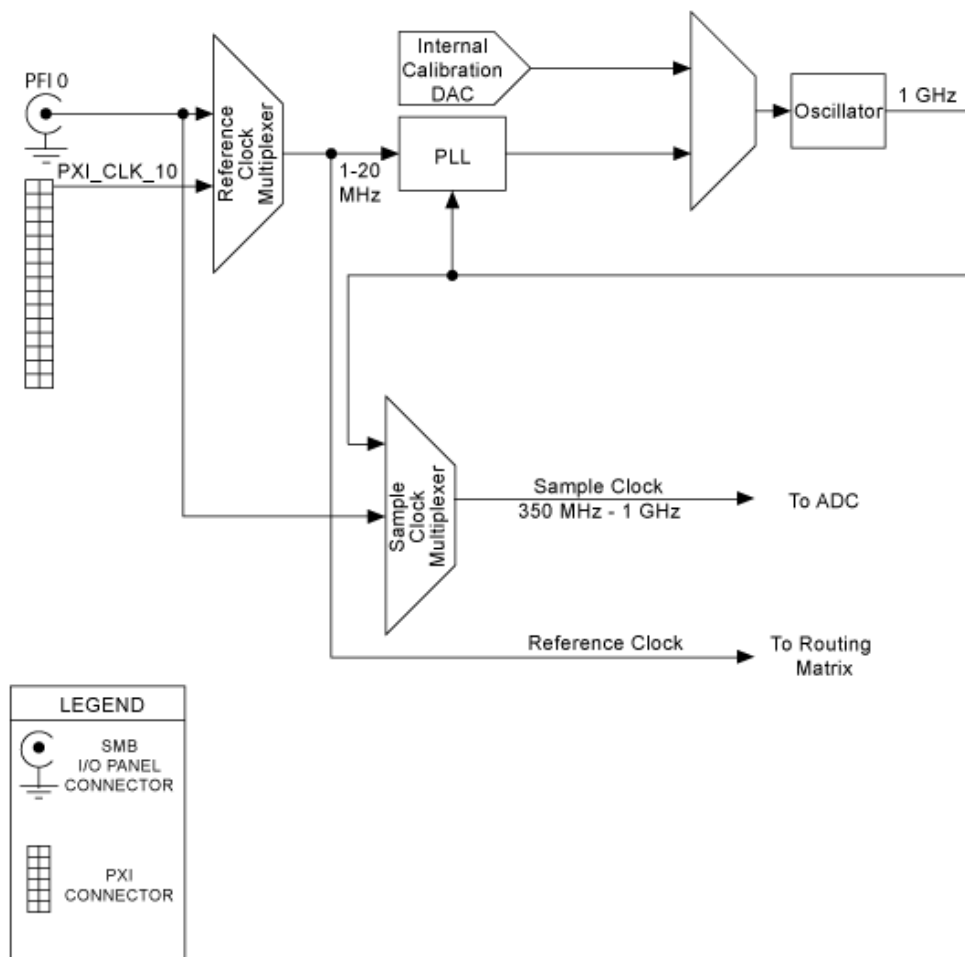
When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI PXI-5152/5153/5154 Clocking

The clock circuitry on the NI PXI-5152/5153/5154 offers versatile clocking options with its ability to use either the internal 1 GHz sample clock or to accept an external sample clock (PFI 0) that you provide. You can also use the phase-locked loop (PLL) circuit on the digitizer to phase lock the internal 1 GHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXI-5152/5153/5154.



Sample Clock

The NI PXI-5152/5153/5154 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PXI-5152/5153/5154 has an onboard voltage controlled SAW oscillator (VCSSO) running at 1 GHz. When using the onboard 1 GHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 1 GHz frequency of the VCSSO. In PLL mode, the device phase locks its 1 GHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 1 GHz clock. In these cases the digitizer can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer at the front panel connector (PFI 0). Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The [reference clock](#) is used in the NI PXI-5152/5153/5154 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizer can accept a reference clock from its front panel (PFI 0) as well as from PXI_CLK10. This reference clock can be any frequency from 1 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



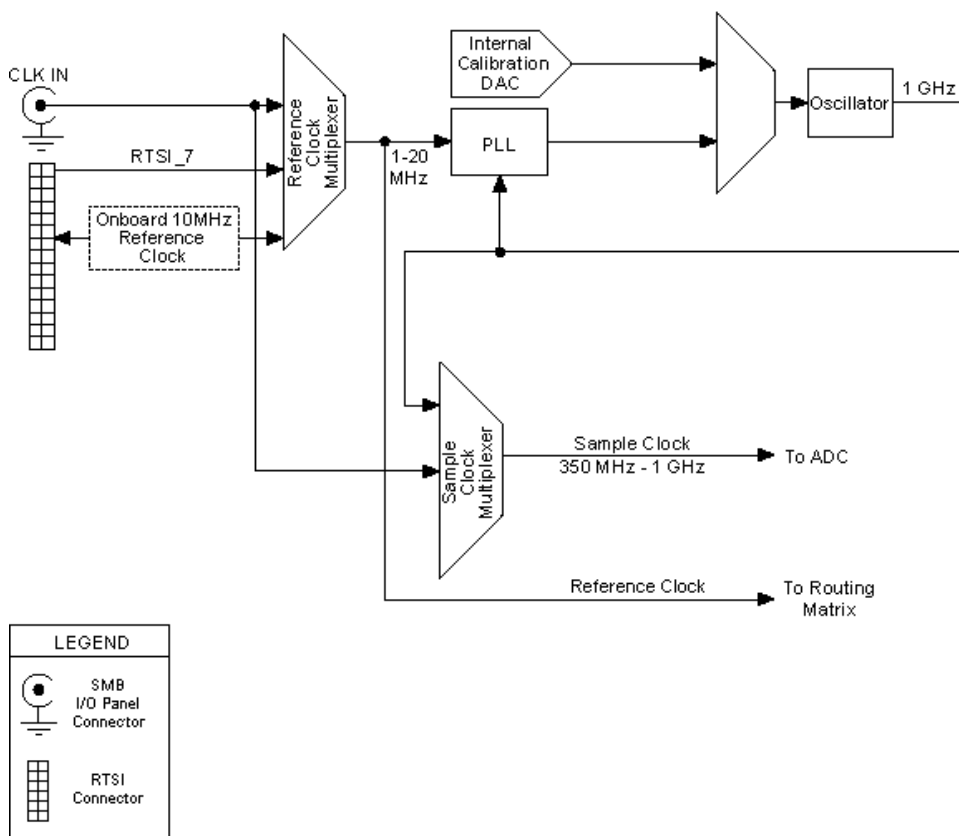
Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock out the PFI 1 connector for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXI-5152/5153/5154 Routing Matrix](#).

NI PCI-5152/5153/5154 Clocking

The clock circuitry on the NI PCI-5152/5153/5154 offers versatile clocking options with its ability to use either the internal 1 GHz sample clock or to accept an external sample clock (PFI 0) that you provide. You can also use the phase-locked loop (PLL) circuit on the digitizer to phase lock the internal 1 GHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5152/5153/5154.



Sample Clock

The NI PCI-5152/5153/5154 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5152/5153/5154 has an onboard voltage controlled SAW oscillator (VCSO) running at 1 GHz. When using the onboard 1 GHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 1 GHz frequency of the VCSO. In PLL mode, the device [phase locks](#) its 1 GHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 1 GHz clock. In these cases the digitizer can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer at the front panel connector (PFI 0). Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

[Reference Clock](#)

The [reference clock](#) is used in the NI PCI-5152/5153/5154 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizer can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 1 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN or RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock out the PFI 1 connector for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5152/5153/5154 Routing Matrix](#).

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5152/5153/5154 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV}$ for temperature range 18-28 °C

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired

temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 \text{ } ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5152/5153/5154, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges and all filter paths (enabled/disabled).
- (NI 5152) Calibrates trigger timing to ensure accurate trigger timing and time-stamping.
- Takes approximately 2-3 minutes (NI 5152) or 5 minutes (NI 5153/5154) to complete.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5152/5153/5154 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

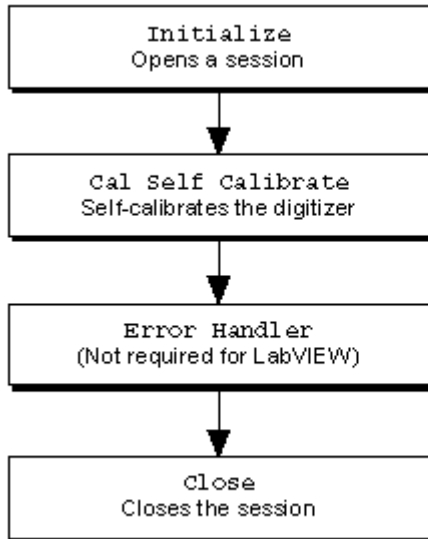
Input Connections During Self-Calibration

The NI 5152/5153/5154 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at Start»All Programs»National Instruments»NI-SCOPE»Examples.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference and TRIG channel	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger timing Trigger level	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

NI 5153/5154

Features

The NI 5153/5154 has the following features:

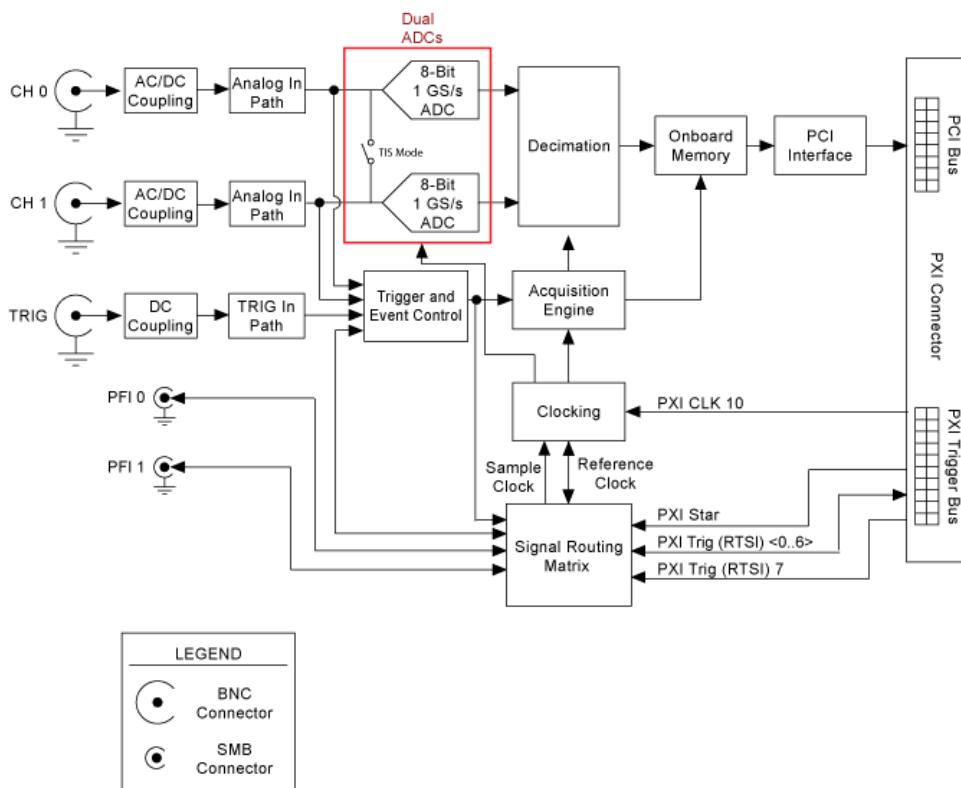
- 2 channels, simultaneously sampled
- 8-bit vertical resolution
- 1 GS/s real-time sampling rate, 2 GS/s time interleaved sampling rate
- Bandwidth
 - NI 5153: 500 MHz
 - NI 5154: 1 GHz
- PXI or PCI versions
- 8 MB, 64 MB, or 256 MB of memory per channel
- NI-TClk synchronization

Related Topics

Features Supported by Device

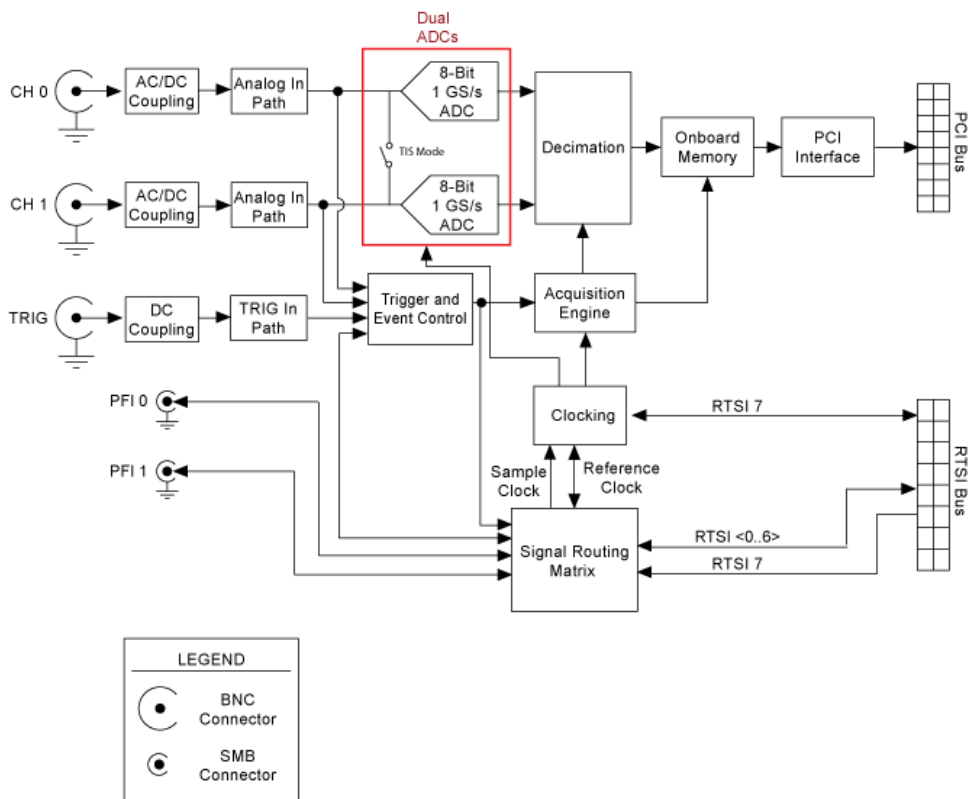
NI PXI-5153/5154 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5153/5154. The dual ADC architecture enables the Time Interleaved Sampling (TIS) feature of these digitizers.



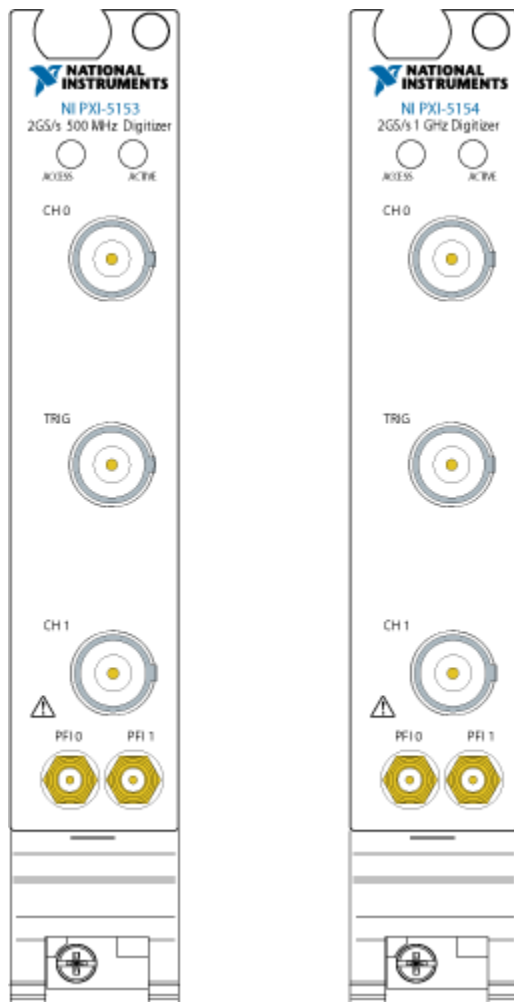
NI PCI-5153/5154 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5153/5154. The dual ADC architecture enables the Time Interleaved Sampling (TIS) feature of these digitizers.



PXI-5153/5154 Front Panel

The following figure shows the front panel of the PXI-5153/5154. Descriptions of the LEDs and connectors are shown below.



LEDs

The PXI-5153/5154 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—The module is unable to detect the external sample clock. ▪ Overvoltage error—The module has detected a 50 Ω overvoltage error. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>

Connectors

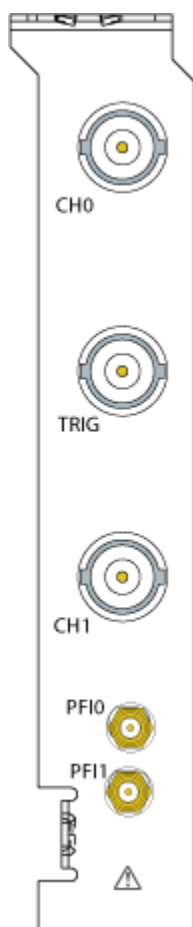
The PXI-5153/5154 has the following connectors on the front panel.

Connector	Description	Function
-----------	-------------	----------

CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
PFI 0	Standard SMB jack	Reference clock in, sample clock in, digital trigger in/out
PFI 1	Standard SMB jack	Reference clock out, +5 V out, digital trigger in/out

PCI-5153/5154 Front Panel

The following figure shows the front panel of the NI PCI-5153/5154. Descriptions of the connectors are shown below.



Connectors

The PCI-5153/5154 has the following connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
PFI 0	Standard SMB jack	Reference clock in, sample clock in, digital trigger in/out
PFI 1	Standard SMB jack	Reference clock out, +5 V out, digital trigger in/out

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.



Notes

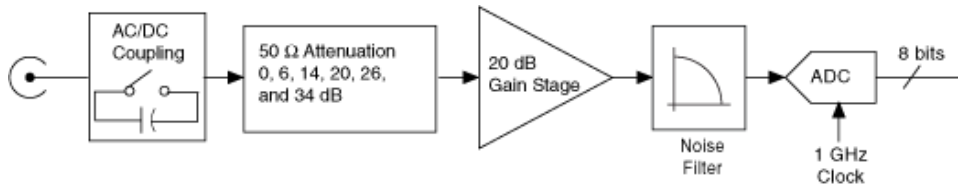
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.


Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5153/5154 Input Signal Conditioning

The NI 5153/5154 provides the following input channel signal conditioning paths with 50 Ω input impedance.



 **Note** The ground on the device inputs is connected to the chassis ground.

NI 5153/5154 Input Ranges

The NI 5153-5154 supports the following input ranges: 0.1, 0.2, 0.5, 1, 2, and 5 V_{pk-pk}

NI 5153/5154 Input Impedance

The NI 5153/5154 has 50 Ω analog input impedance.

Protection

The 50 Ω input of the NI 5153/5154 is protected by a voltage sense circuit. If an overvoltage event is large and sudden enough, however, the protection circuits might not have enough time to react before permanent damage occurs. It is therefore important that you observe the specified maximum signal input levels.

NI 5153/5154 AC/DC Coupling

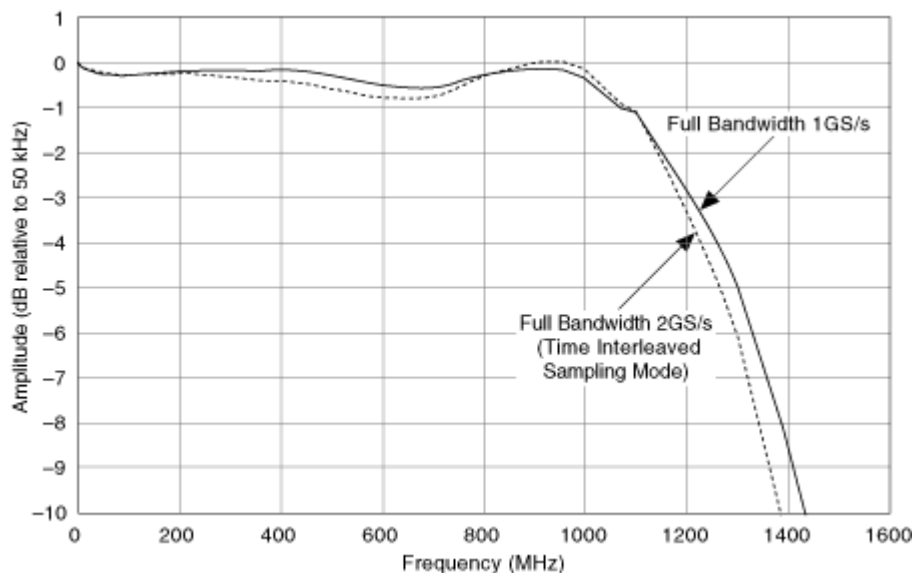
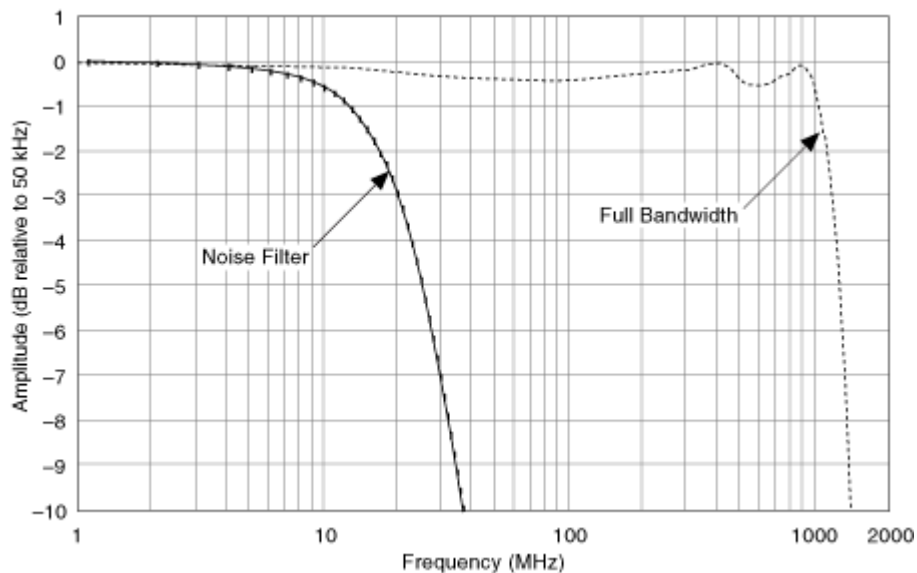
You can select AC, DC, or GND [input coupling](#) for the 50 Ω input path. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. A DC input offset adjustment is available if the signal you want to measure is below this limit.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5153/5154 Noise Filter

The NI 5153/5154 provides a 20 MHz noise filter that limits the bandwidth of the signal path through the 50 Ω signal path. This filter is intended to reduce noise when the signal content is 20 MHz or less.

Typical frequency responses of the NI 5154 are shown in the following figures.



Note For more information about NI 5153 frequency response, refer to the specifications document that shipped with the device.

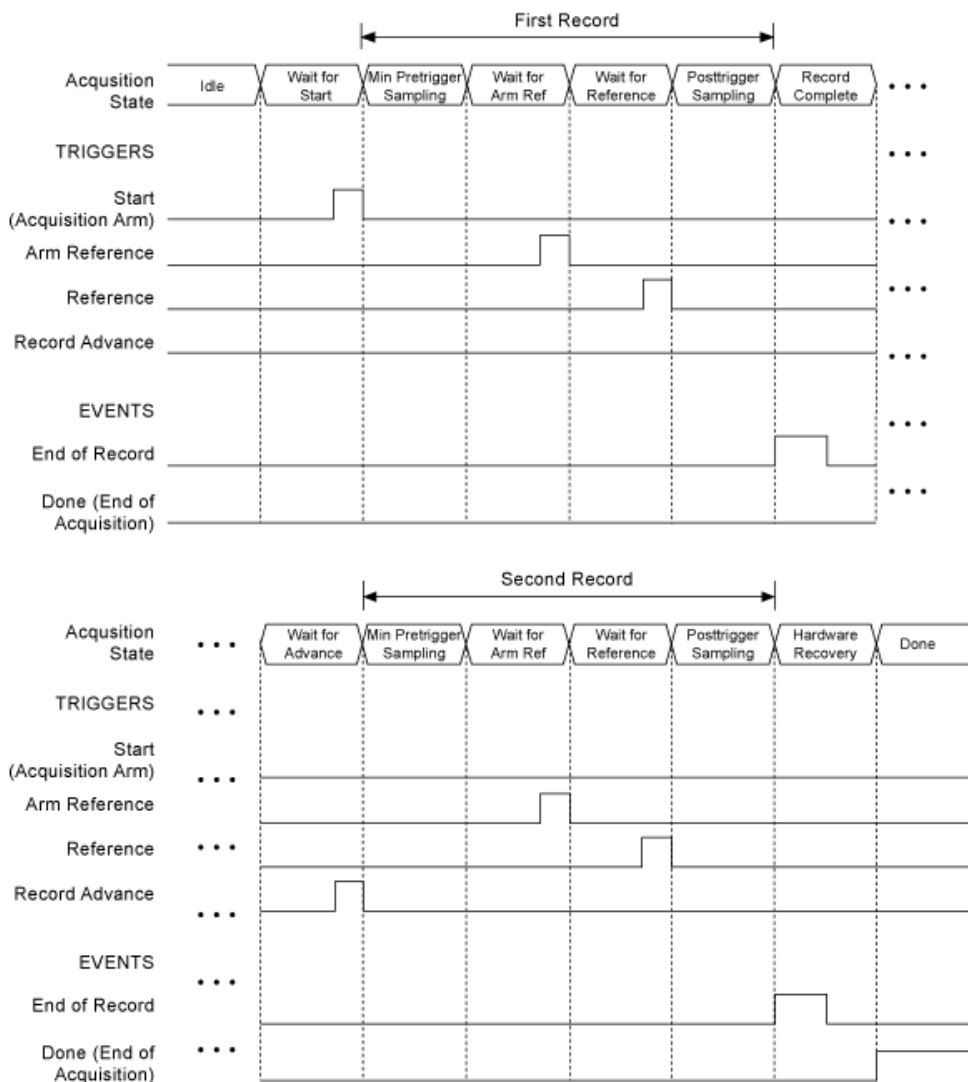
SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI

5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins

storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5152/5153/5154 Routing Matrix

The following table shows the signals available for export from the NI PXI-5152/5153/5154 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	PFI 0	PFI 1
Exported Clocks			
Reference Clock (External)	√	—	√
Triggers			
Acquisition Arm (Start) Trigger	√	√	√
Reference (Stop) Trigger	√	√	√
Events			
End of Record Event	√	√	√
End of Acquisition Event	√	√	√
Ready for Start Event	√	√	√
Ready for Reference Event	√	√	√
Ready for Advance Event	√	√	√

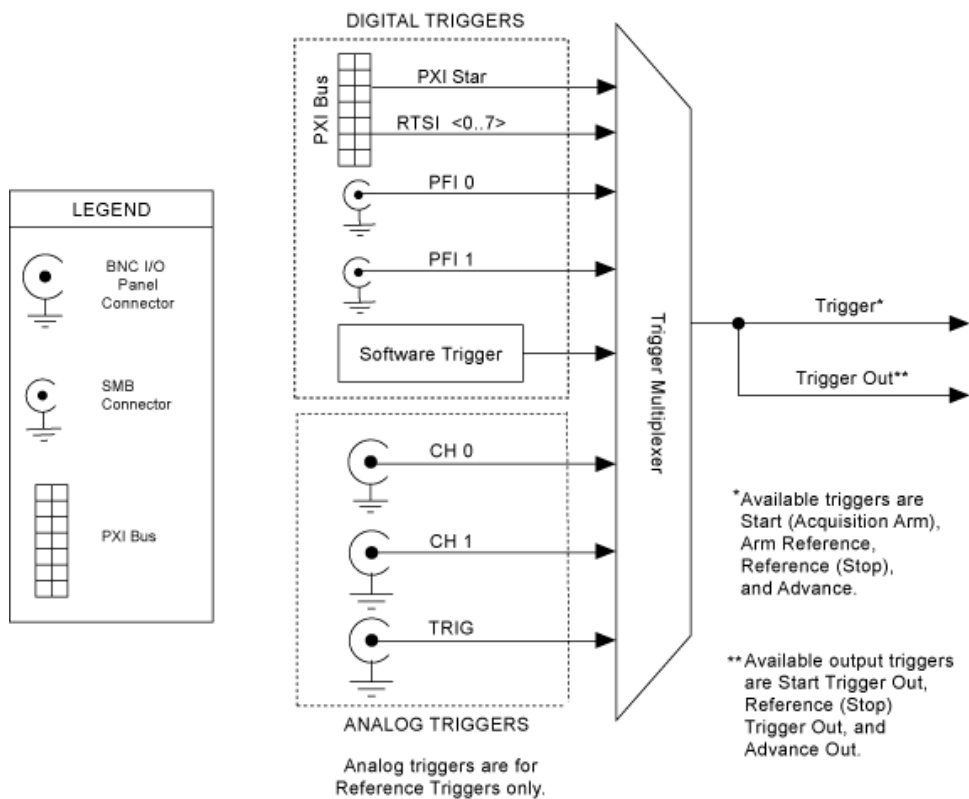
NI PCI-5152/5153/5154 Routing Matrix

The following table shows the signals available for export from the NI PCI-5152/5153/5154 and the lines to which they can be routed.

Source	Destination			
	RTSI <0..6> (RTSI Bus)	PFI 0	PFI 1	RTSI 7
Exported Clocks				
Reference Clock (External)	√	—	√	—
Reference Clock (Internal)	—	√	√	√
Triggers				
Acquisition Arm (Start) Trigger	√	√	√	—
Reference (Stop) Trigger	√	√	√	—
Events				
End of Record Event	√	√	√	—
End of Acquisition Event	√	√	√	—
Ready for Start Event	√	√	√	—
Ready for Reference Event	√	√	√	—
Ready for Advance Event	√	√	√	—

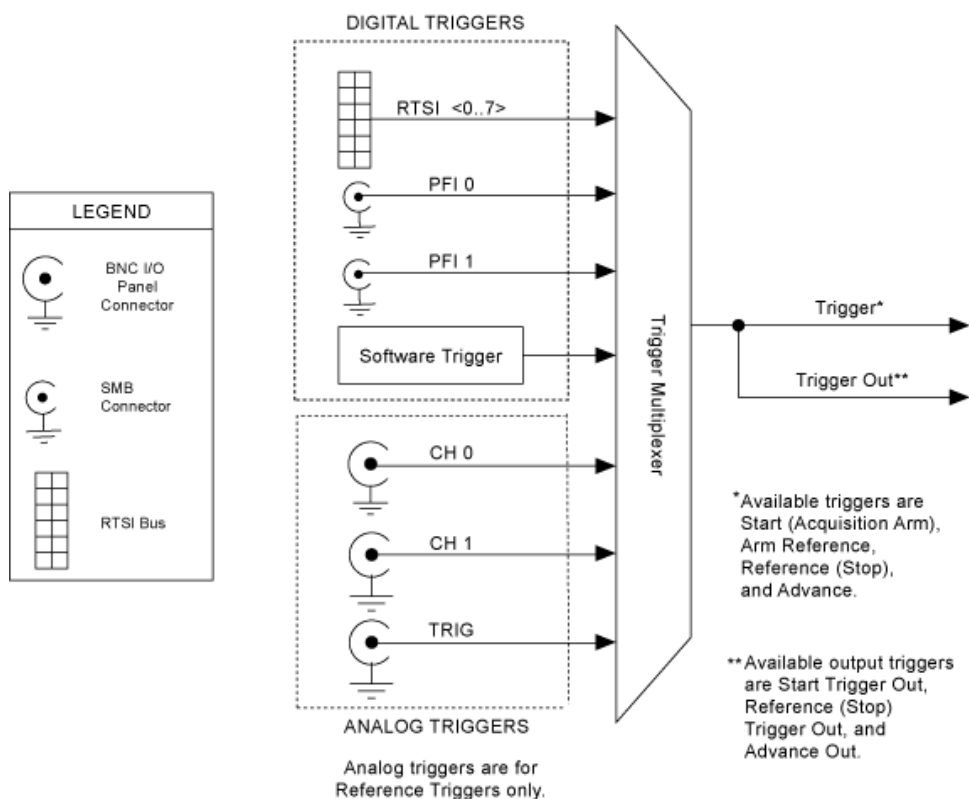
NI PXI-5152/5153/5154 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5152/5153/5154.



NI PCI-5152/5153/5154 Trigger Sources


The following figure shows the trigger sources for the NI PCI-5152/5153/5154.



NI 5114/5152/5153/5154 Trigger Delay

Trigger delay, which is specified in seconds, is achieved on the NI 5114/5152/5153/5154 by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

$$\text{Max trigger delay in seconds} = [(2^{35} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$

 **Note** The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock.

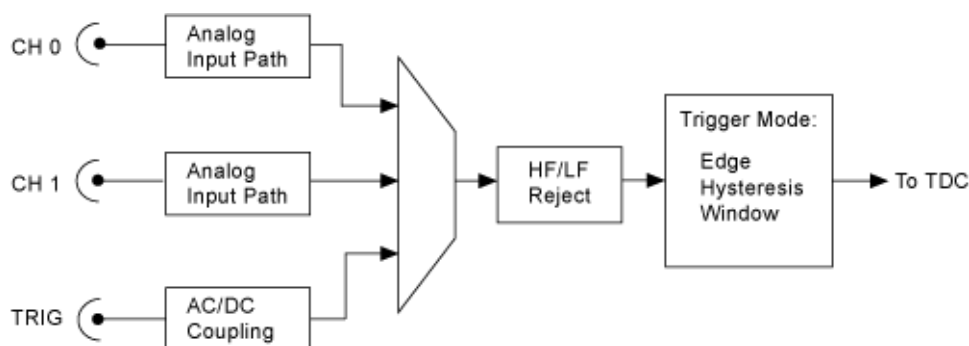
NI 5152/5153/5154 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5152/5153/5154 provides flexible, high-precision, low-jitter triggering features.

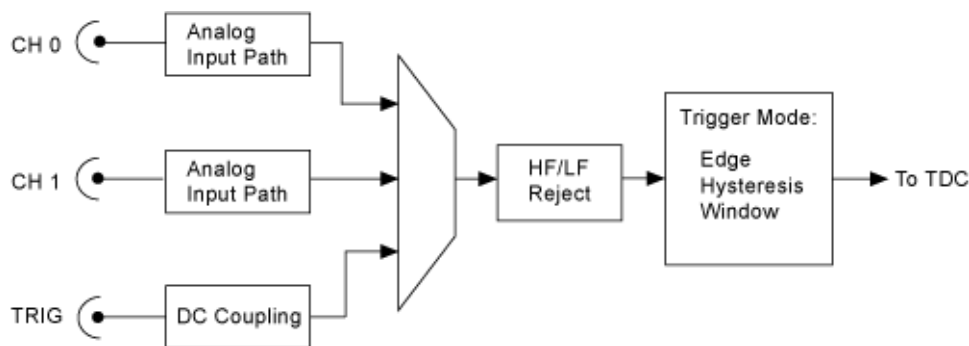
The NI 5152/5153/5154 has three fundamental analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5152.



The following figure shows the analog trigger paths for the NI 5153/5154.



External Trigger Channel (TRIG)

Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range and impedance to give a higher bandwidth for triggering the digitizer. Signals that travel through the external trigger channel are not digitized.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Filters

LF and HF Reject Filters

The NI 5114/5122/5124/5142/5152/5153/5154 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 50 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 50 kHz highpass filter, the 50 or 60 Hz component is attenuated by nearly 60 dB, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 50 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5114/5122/5124/5142/5152/5153/5154 Trigger Holdoff

For NI 5114/5122/5124/5142/5152/5153/5154 devices, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

TDC On

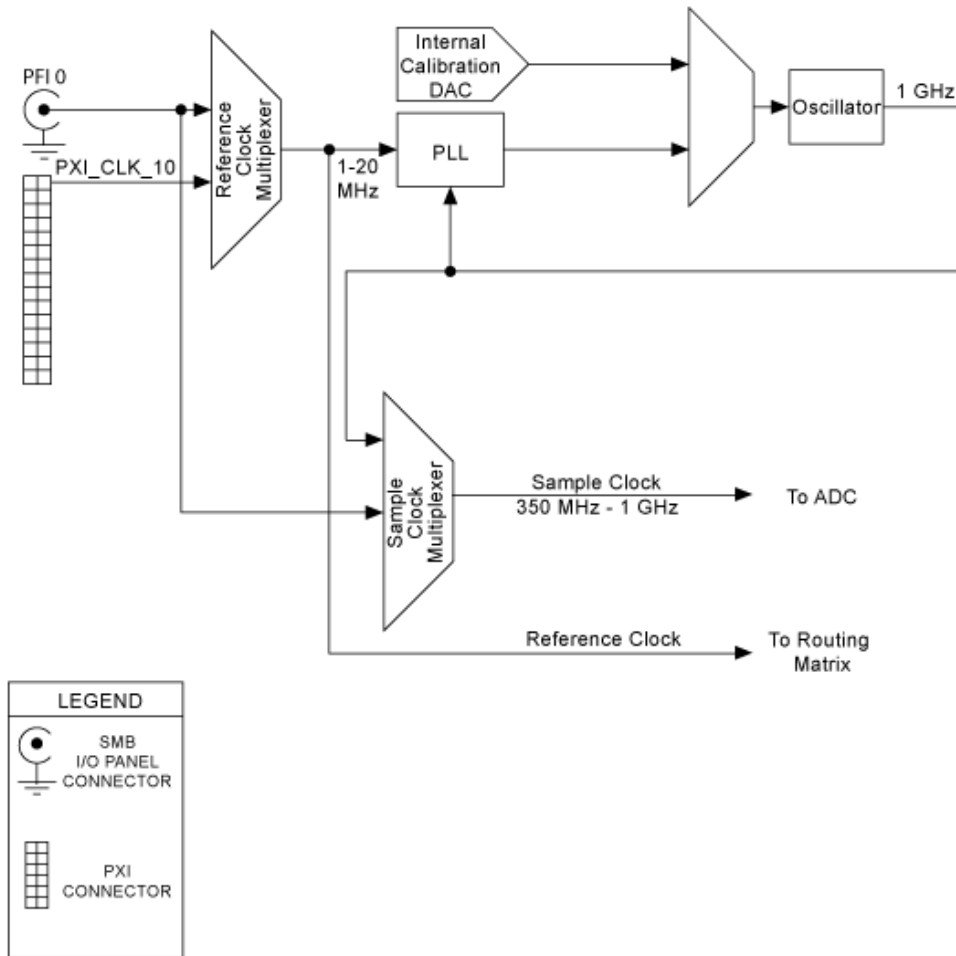
When the [time-to-digital converter \(TDC\)](#) is enabled on the digitizer, the minimum holdoff you can set is 10 μs (for the NI 5114/5122/5124/5142) or 8 μs (for the NI 5152/5153/5154). This minimum holdoff time between Reference Triggers is required for the TDC to settle; any holdoff value below these values is coerced up.

TDC Off

When the TDC is disabled, the minimum holdoff value you can set is decreased to 2 μs (for the NI 5114/5122/5124/5142) or 1 μs (for the NI 5152/5153/5154) when using the internal sample clock. For minimum trigger holdoff when using an external sample clock, refer to the specifications document for your digitizer.

NI PXI-5152/5153/5154 Clocking

The clock circuitry on the NI PXI-5152/5153/5154 offers versatile clocking options with its ability to use either the internal 1 GHz sample clock or to accept an external sample clock (PFI 0) that you provide. You can also use the phase-locked loop (PLL) circuit on the digitizer to phase lock the internal 1 GHz sample clock with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXI-5152/5153/5154.



Sample Clock

The NI PXI-5152/5153/5154 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PXI-5152/5153/5154 has an onboard voltage controlled SAW oscillator (VCSO) running at 1 GHz. When using the onboard 1 GHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 1 GHz frequency of the VCSO. In PLL mode, the device phase locks its 1 GHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 1 GHz clock. In these cases the digitizer can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer at the front panel connector (PFI 0). Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

Reference Clock

The [reference clock](#) is used in the NI PXI-5152/5153/5154 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizer can accept a reference clock from its front panel (PFI 0) as well as from PXI_CLK10. This reference clock can be any frequency from 1 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



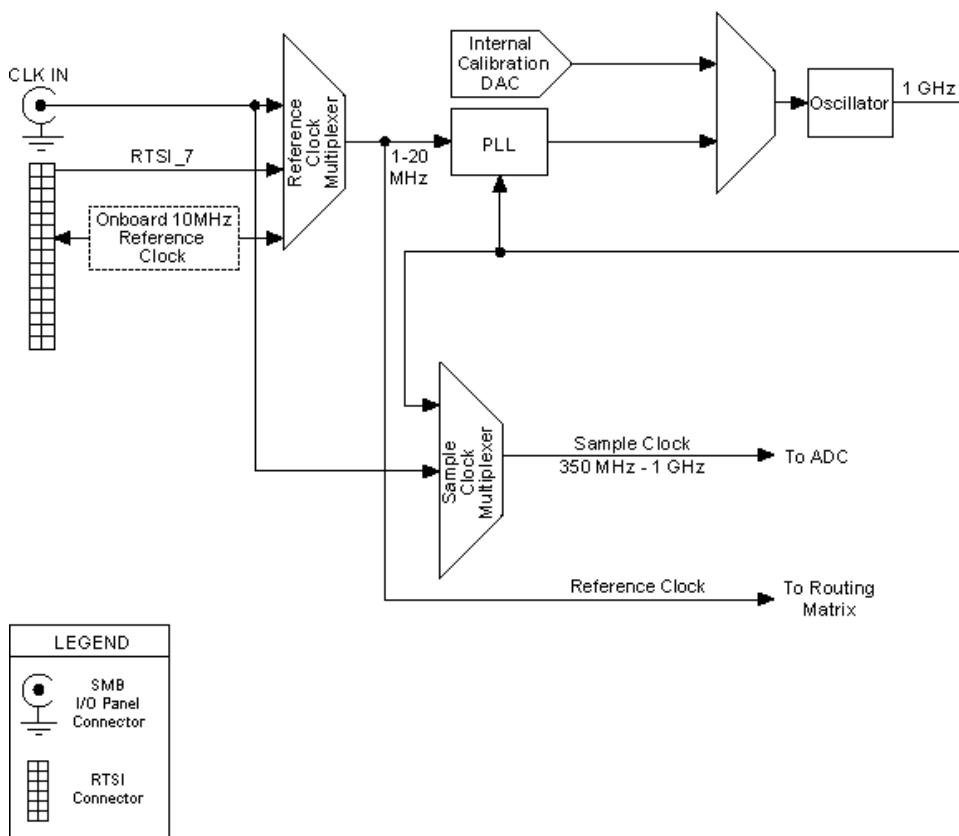
Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock out the PFI 1 connector for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXI-5152/5153/5154 Routing Matrix](#).

NI PCI-5152/5153/5154 Clocking

The clock circuitry on the NI PCI-5152/5153/5154 offers versatile clocking options with its ability to use either the internal 1 GHz sample clock or to accept an external sample clock (PFI 0) that you provide. You can also use the phase-locked loop (PLL) circuit on the digitizer to phase lock the internal 1 GHz sample clock with the internal 10 MHz reference, a reference clock from another module on RTSI 7 (RTSI Clock), or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5152/5153/5154.



Sample Clock

The NI PCI-5152/5153/5154 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI PCI-5152/5153/5154 has an onboard voltage controlled SAW oscillator (VCSO) running at 1 GHz. When using the onboard 1 GHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 1 GHz frequency of the VCSO. In PLL mode, the device [phase locks](#) its 1 GHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 1 GHz clock. In these cases the digitizer can accept an external sample clock. External clocking also provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer at the front panel connector (PFI 0). Refer to the hardware [specifications document](#) for external sample clock requirements.



Note When you use an external sample clock, the [TDC](#) is disabled. Refer to the hardware [specifications document](#) for information on how this affects [trigger holdoff](#) and trigger resolution.

[Reference Clock](#)

The [reference clock](#) is used in the NI PCI-5152/5153/5154 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizer can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 1 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN or RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock out the PFI 1 connector for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5152/5153/5154 Routing Matrix](#).

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5152/5153/5154 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV}$ for temperature range 18-28 °C

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired

temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$TC = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 ^\circ\text{C} \times TC = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5152/5153/5154, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges and all filter paths (enabled/disabled).
- (NI 5152) Calibrates trigger timing to ensure accurate trigger timing and time-stamping.
- Takes approximately 2-3 minutes (NI 5152) or 5 minutes (NI 5153/5154) to complete.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5152/5153/5154 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

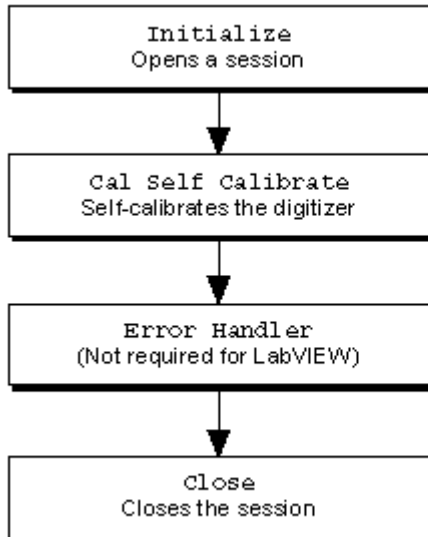
Input Connections During Self-Calibration

The NI 5152/5153/5154 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/ μ s slew rate) are present during self-calibration.

When in doubt, disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/ μ s, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at Start»All Programs»National Instruments»NI-SCOPE»Examples.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference and T RIG channel	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger timing Trigger level	90 days, or when temperature changes >5 °C	Ensures range to range matching Ensures trigger accuracy
No calibration	None, within 2 year calibration cycle or if temperature stays within ±5 °C	High accuracy not required outside of 5 °C	If self-calibration is not used, derate the accuracy using the specified Temperature

PXIe-5160/5162

The PXIe-5160/5162 has the following features:

- 2 or 4 input channels, simultaneously sampled
- 10-bit vertical resolution
- Maximum sample rate (with internal sample clock; based on the channels enabled)

Number of Input Channels	Channels Enabled	Maximum Sample Rate	
		PXIe-5160	PXIe-5162
2	1	2.5 GS/s	5 GS/s
	2	2.5 GS/s	
4	1	2.5 GS/s	5 GS/s
	CH 0 and CH 2	2.5 GS/s	
	Any other 2-channel combination	1.25 GS/s	
	≥3	1.25 GS/s	

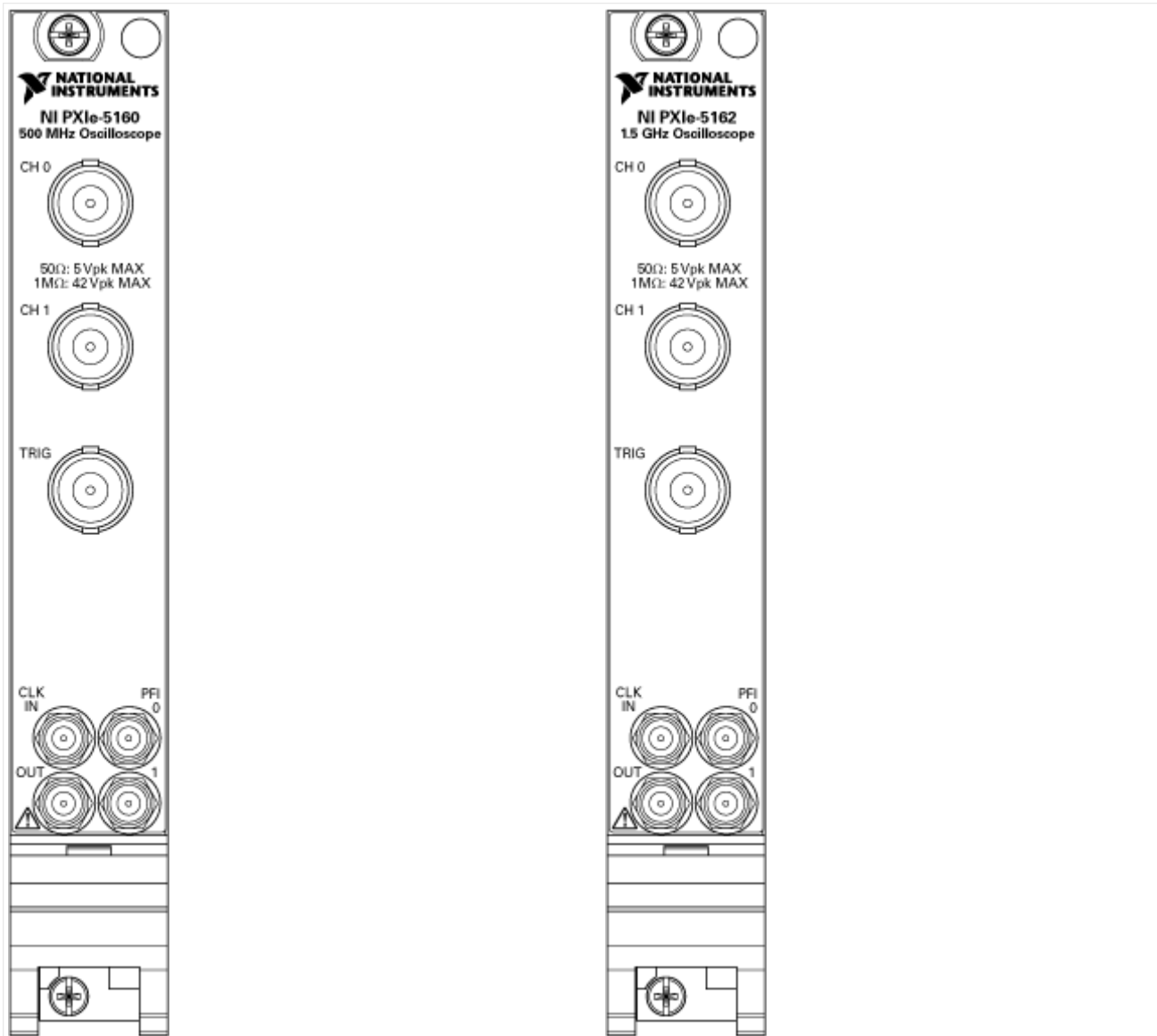
- Gen 1 x4 PXI Express
- Up to 2 GB onboard memory
- Bandwidth
 - PXIe-5160: 500 MHz (50 Ω), 300 MHz (1 M Ω)
 - PXIe-5162: 1.5 GHz (50 Ω), 300 MHz (1 M Ω)
- 20 MHz and 175 MHz filters
- AC and DC coupling
- Triggers
 - PXIe-5160/5162 (2 CH): Analog edge trigger and external trigger with sub-sample clock accuracy
 - PXIe-5160/5162 (4 CH): Analog edge trigger with sub-sample clock accuracy
- [NI-TClk synchronization](#)
- Peer-to-peer streaming

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5160/5162 (2 CH) Front Panel

The following figure shows the front panel of the NI 5160/5162 (2 CH) device. Descriptions of the connectors are shown below.



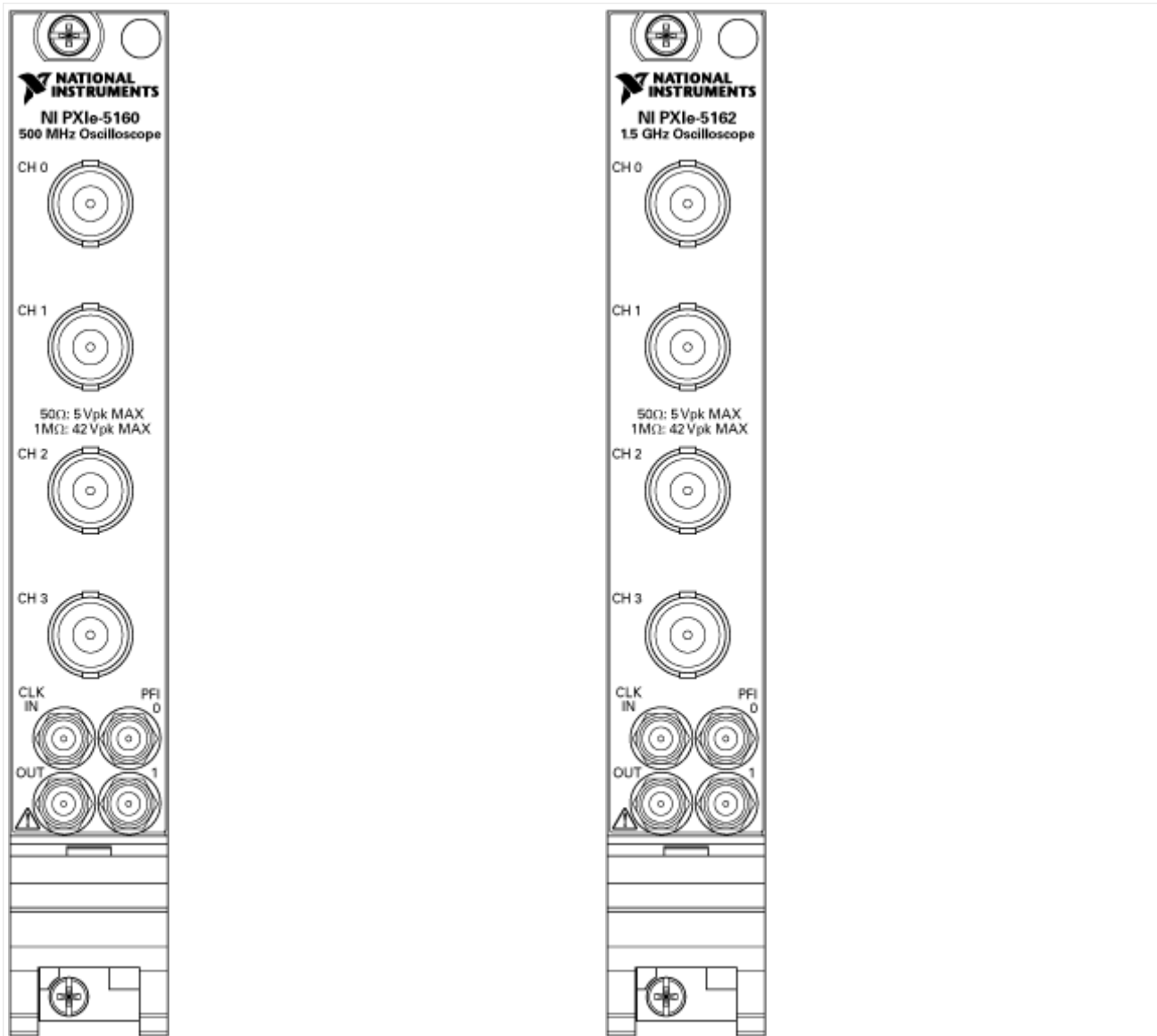
Connectors

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions.
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized.

CLK IN	Standard SMB jack	Reference clock in, sample clock in.
CLK OUT	Standard SMB jack	Reference clock out.
PFI 0	Standard SMB jack	Digital trigger in/out.
PFI 1	Standard SMB jack	Digital trigger in/out, no sub-sample trigger accuracy, probe compensation.

NI 5160/5162 (4 CH) Front Panel

The following figure shows the front panel of the NI 5160/5162 (4 CH) device. Descriptions of the connectors are shown below.



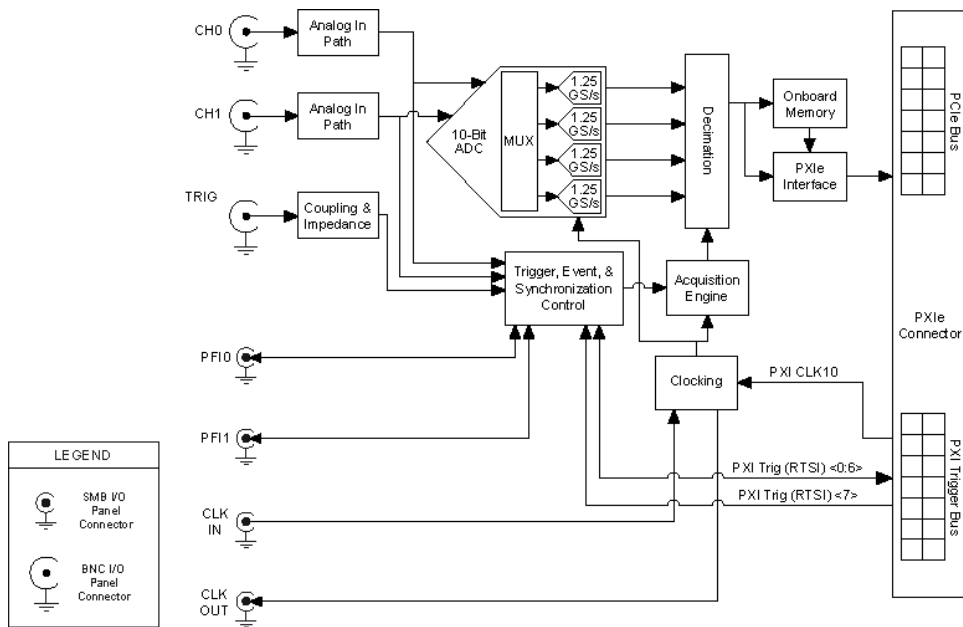
Connectors

Connector	Description	Function
CH 0...CH 3	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions.
CLK IN	Standard SMB jack	Reference clock in, sample clock in.
CLK OUT	Standard SMB jack	Reference clock out.
PFI 0	Standard SMB jack	Digital trigger in/out.

PFI 1	Standard SMB jack	Digital trigger in/out, no sub-sample trigger accuracy, probe compensation.
-------	-------------------	---

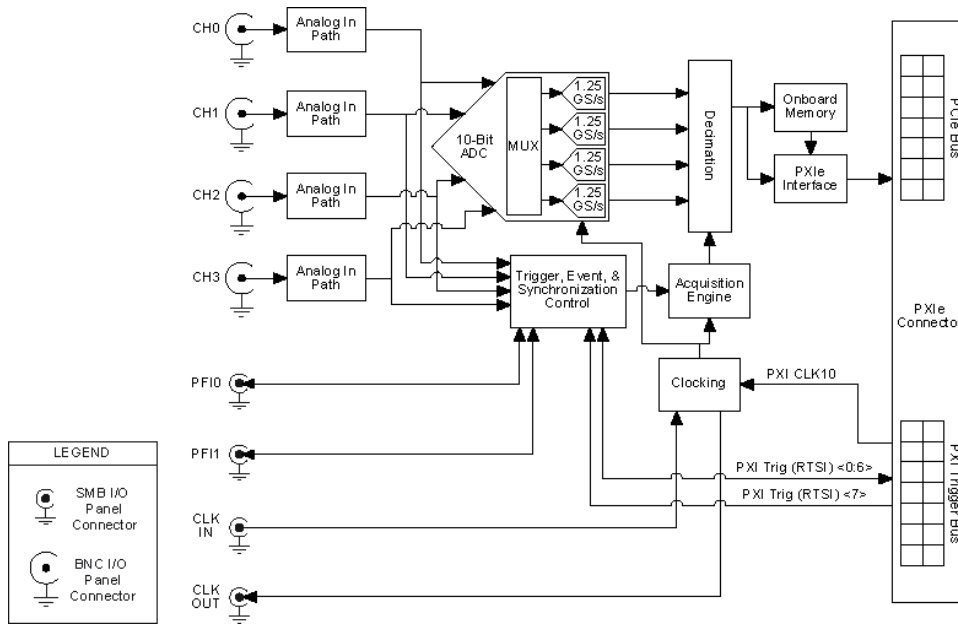
NI 5160/5162 (2 CH) Block Diagram

The following figure shows a detailed block diagram of the NI 5160/5162 (2 CH).



NI 5160/5162 (4 CH) Block Diagram

The following figure shows a detailed block diagram of the NI 5160/5162 (4 CH).



Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.



Notes

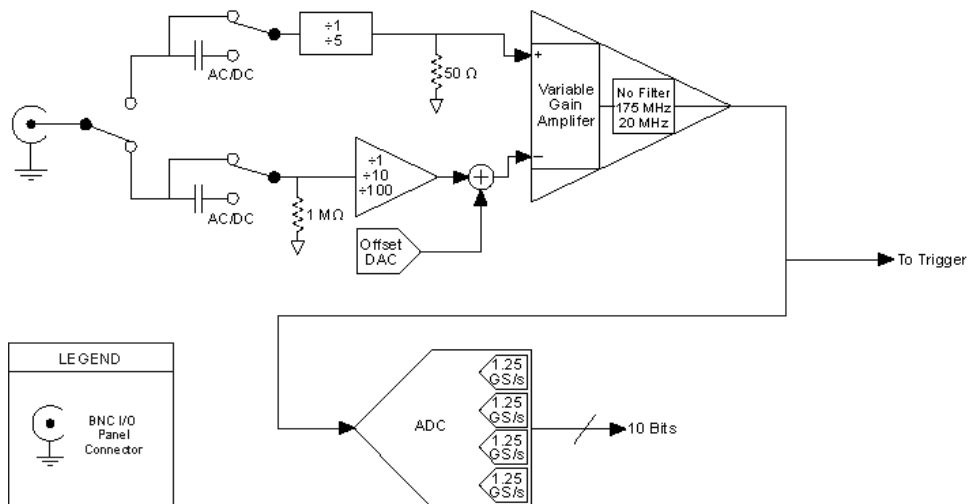
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5160/5162 Input Signal Conditioning

The NI 5160/5162 provides the following input channel signal conditioning path.



Note The ground on the device inputs is connected to the chassis ground.

NI 5160/5162 Input Ranges

As shown in the following table, the 1 MΩ path allows up to 50 V_{pk-pk} whereas the 50 Ω path allows up to 5 V_{pk-pk}.

50Ω Input Path	1 MΩ Input Path
0.05 V _{pk-pk}	0.05 V _{pk-pk}
0.1 V _{pk-pk}	0.1 V _{pk-pk}
0.2 V _{pk-pk}	0.2 V _{pk-pk}
0.5 V _{pk-pk}	0.5 V _{pk-pk}
1 V _{pk-pk}	1 V _{pk-pk}
2 V _{pk-pk}	2 V _{pk-pk}
5 V _{pk-pk}	5 V _{pk-pk}
—	10 V _{pk-pk}
—	20 V _{pk-pk}
—	50 V _{pk-pk}

Note Vertical offset can be set to effectively create a unipolar range. Refer to [NI 5160/5162 Vertical Offset](#) for more information.

NI 5160/5162 Input Impedance

You can set the NI 5160/5162 analog input impedance to either 50 Ω or 1 M Ω . The 1 M Ω path is required in applications that require minimal loading or use of a passive oscilloscope probe.

Protection

The 50 Ω inputs of the NI 5160/5162 are protected by a voltage sense circuit. If an overvoltage event is large and sudden enough, the protection circuits might not have enough time to react before permanent damage occurs. It is important that you observe the specified maximum signal input levels, especially when the inputs are set for 50 Ω .

NI 5160/5162 Coupling

You can select AC or DC [input coupling](#) for the 50 Ω and 1 M Ω input paths. Select AC-coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency bandpass flatness. If the signal you want to measure is below the limit of the AC coupling cutoff frequency, a DC input offset adjustment is available. For information about the AC coupling cutoff frequency, refer to the **NI 5160 Specifications** or the **NI 5162 Specifications**.

NI 5160/5162 Vertical Offset

The following table shows the maximum valid vertical offset for each range on the NI 5160/5162.

Range	50 Ω Vertical Offset	1 M Ω Vertical Offset
0.05 V _{pk-pk}	± 0.5 V	± 0.5 V
0.1 V _{pk-pk}	± 0.5 V	± 0.5 V
0.2 V _{pk-pk}	± 0.5 V	± 0.5 V
0.5 V _{pk-pk}	± 0.5 V	± 0.5 V
1 V _{pk-pk}	± 0.5 V	± 0.5 V
2 V _{pk-pk}	± 1.5 V	± 5 V

5 V _{pk-pk}	0 V	±5 V
10 V _{pk-pk}	—	±5 V
20 V _{pk-pk}	—	±30 V
50 V _{pk-pk}	—	±15 V

On the NI 5160/5162, you can set a hardware-applied vertical offset to effectively create a unipolar range. For example, if you set the vertical offset to -1 V for the 2 V_{pk-pk} range, you can measure input voltages between 0 and -2 V.

NI 5160/5162 Filters

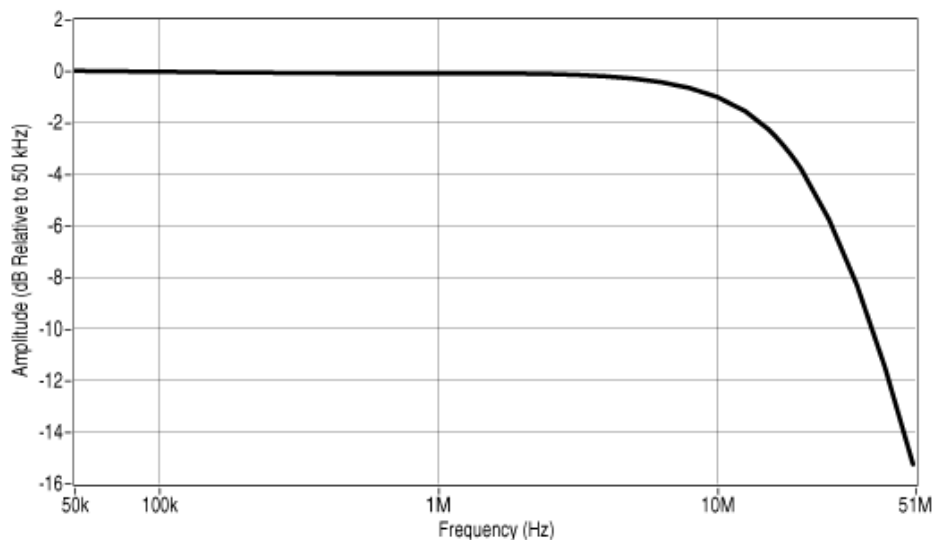
The NI 5160/5162 offers two filters that limit the bandwidth of the signal path: a 20 MHz noise filter and a 175 MHz noise filter. Both filters are available through the 1 M Ω and 50 Ω signal paths. The 20 MHz filter is intended to reduce noise when the signal content is 20 MHz or less, and the 175 MHz filter is intended to reduce noise when the signal content is 175 MHz or less.

The noise filter is useful for rejecting out of band or background noise in lower frequency applications. An example application would be characterization of video waveforms in the presence of IF noise. Because most of the signal content is below the 8 MHz range, the 20 MHz noise filter is ideal. Many standard noise tests specify a 20 MHz noise bandwidth.

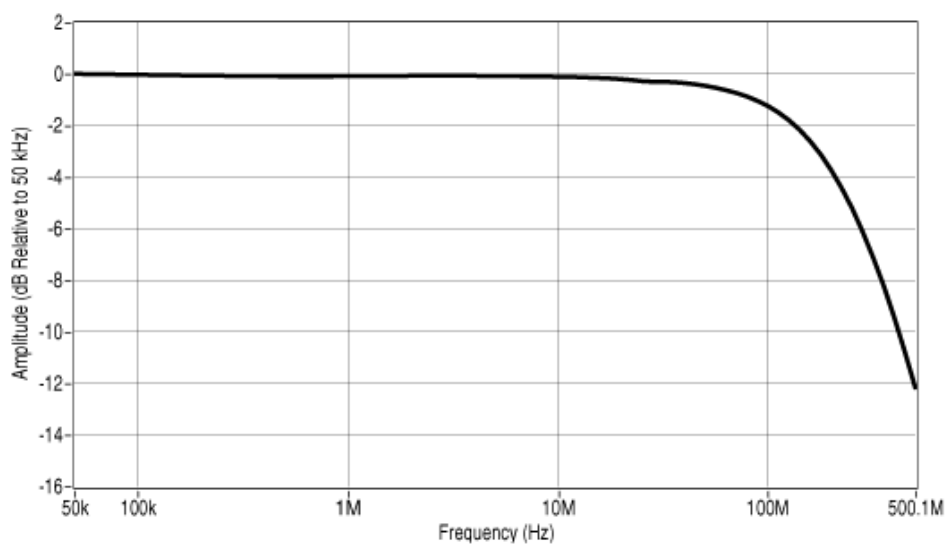


Note To change the filter type, call [niScope Configure Chan Characteristics VI](#) or the [niScope_ConfigureChanCharacteristics](#) function and set the Max Input Frequency parameter to the appropriate value. For example, setting the Max Input Frequency parameter to 8M would enable the 20 MHz filter.

The following figure shows a characteristic frequency response of the 20 MHz noise filter.



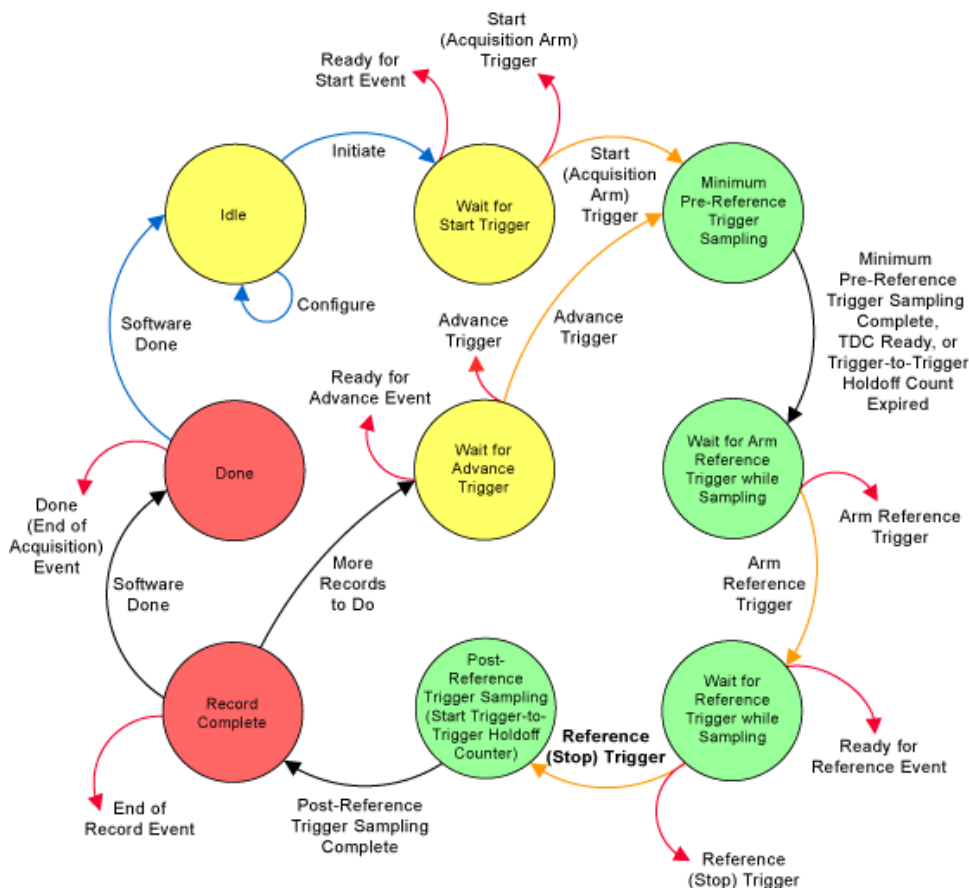
The following figure shows a characteristic frequency response of the 175 MHz noise filter.



Refer to the [hardware specifications](#) document for more information on these filters.

SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

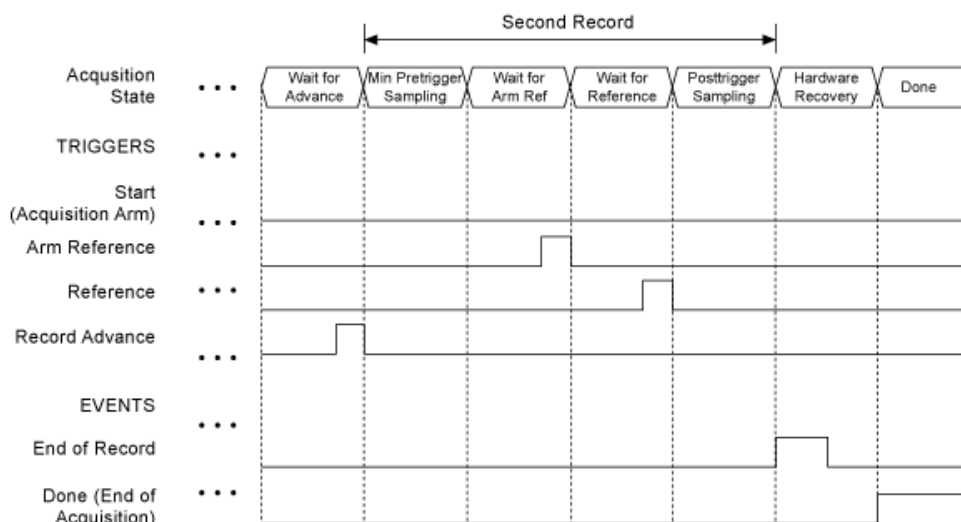
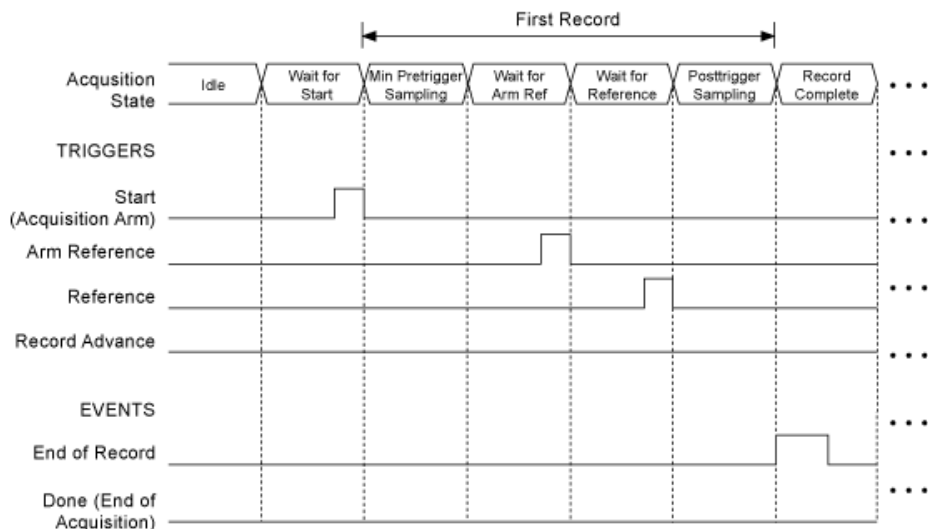
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)


SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.

 **Note** The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI 5160/5162 Routing Matrix

The following table shows the signals available for export from the NI PXIe-5160/5162 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0:6>	CLK OUT	PFI <0:1>
Exported Clocks			
Reference Clock	—	√	—
Sample Clock	—	—	—
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

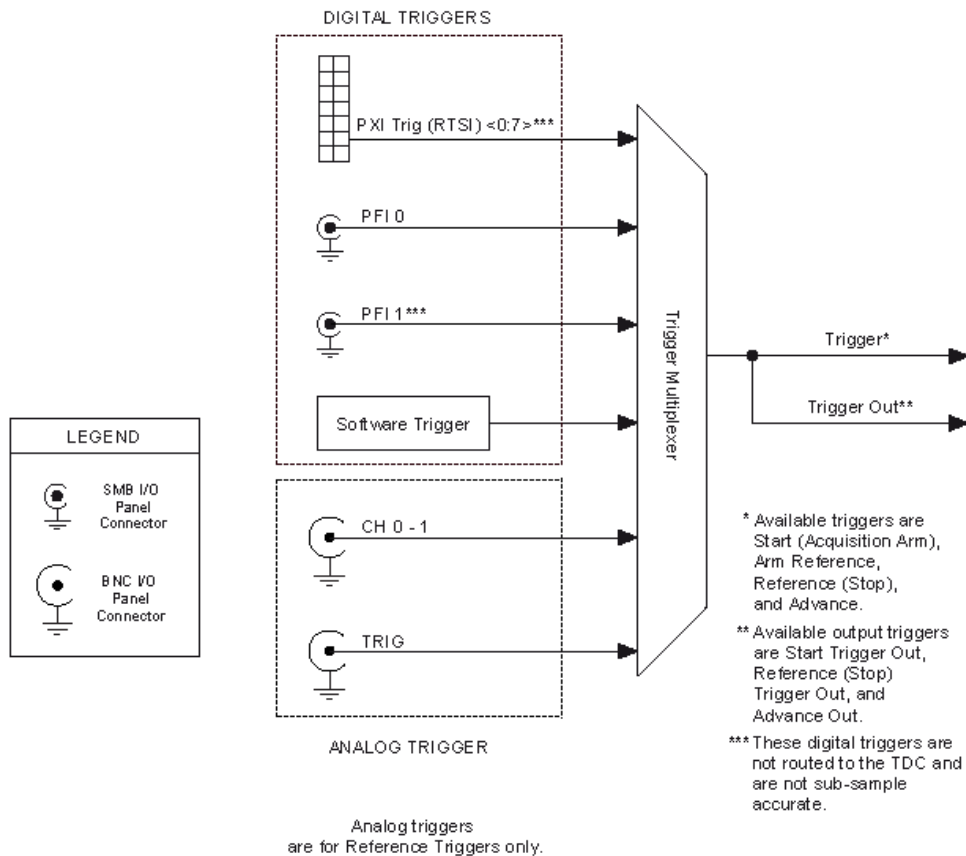
For more information about the routing capabilities of the NI 5160/5162, refer to the [NI 5160/5162 \(2 CH\) Block Diagram](#), [NI 5160/5162 \(4 CH\) Block Diagram](#), and [SMC-Based Digitizers Timing Diagram](#).

NI 5160/5162 Analog Trigger Types

The NI 5160/5162 supports analog [edge](#) and [hysteresis](#) trigger types.

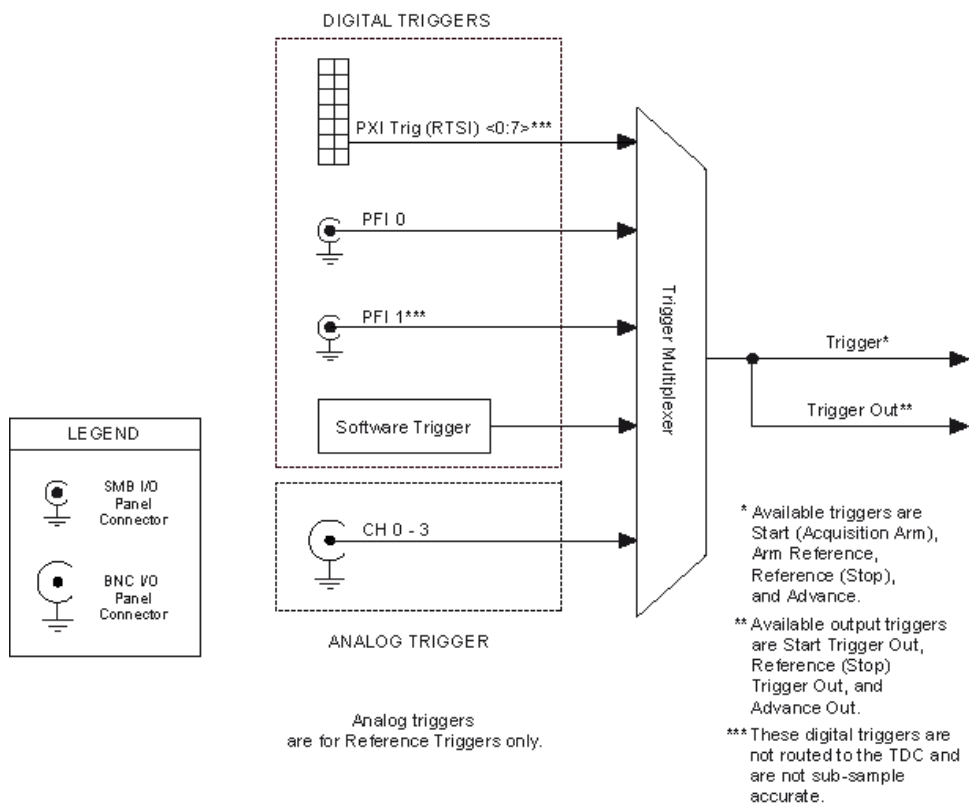
NI 5160/5162 (2 CH) Trigger Sources

The following figure shows the trigger sources for NI 5160/5162 (2 CH) digitizers.



NI 5160/5162 (4 CH) Trigger Sources

The following figure shows the trigger sources for NI 5160/5162 (4 CH) digitizers.



NI 5160/5162 Analog Trigger Paths

With high-speed digitizers, triggering capability is as important as input signal conditioning. The NI 5160/5162 provides flexible, high-precision, low-jitter triggering features.

Trigger Channels

You can use a disabled analog input channel as a trigger. Since the sample rate on the NI 5160/5162 may be limited based on the number of channels you have enabled, you may want to use a channel as a trigger without enabling the channel. Refer to the NI KnowledgeBase for more information.

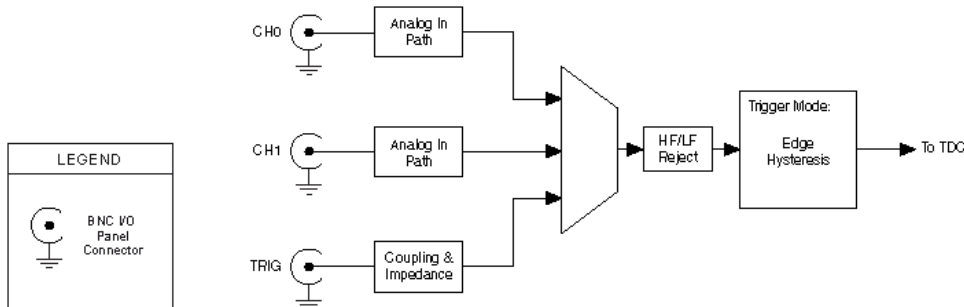
NI 5160/5162 (2 CH) Analog Trigger Paths

NI 5160/5162 (2 CH) digitizers have three analog trigger paths:

- Channel 0 input (CH 0)

- Channel 1 input (CH 1)
- External trigger input (TRIG)

The following figure shows the analog trigger paths for the NI 5160/5162 (2 CH).



External Trigger Channel (TRIG)

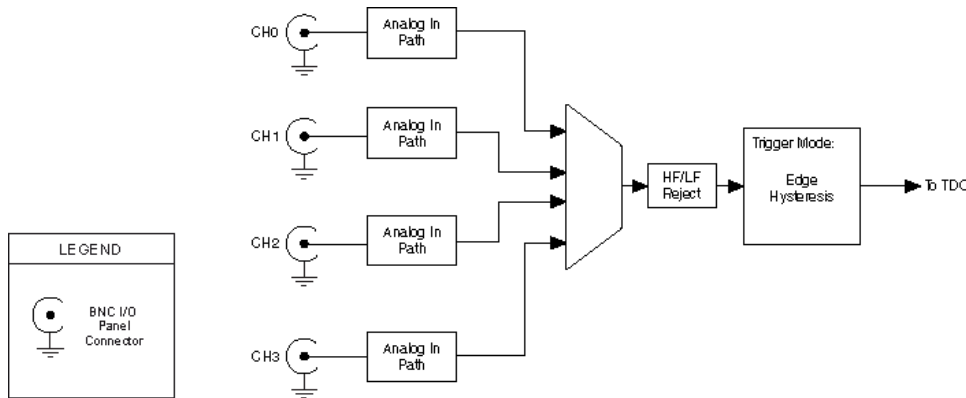
Unlike the input channels (CH 0 and CH 1), the external trigger channel uses a fixed input range of $5 V_{pk-pk}$ for the 50Ω path and $10 V_{pk-pk}$ for the $1 M\Omega$ path. Signals that travel through the external trigger channel are not digitized.

NI 5160/5162 (4 CH) Analog Trigger Paths

NI 5160/5162 (4 CH) digitizers have four analog trigger paths:

- Channel 0 input (CH 0)
- Channel 1 input (CH 1)
- Channel 2 input (CH 2)
- Channel 3 input (CH 3)

The following figure shows the analog trigger paths for the NI 5160/5162 (4 CH).



NI 5160/5162 Trigger Filters

LF and HF Reject Filters

The NI 5160/5162 has two filter selections, LF Reject and HF Reject, available for all analog trigger sources. These are defined as trigger coupling options. Both are single pole filters, with the cutoff frequency set to 150 kHz.

LF Reject Example

An example application for the LF Reject is in triggering on a 455 kHz IF signal in the presence of line cycle noise. If the 50 or 60 Hz line signal has sufficient magnitude, it causes triggering and timing errors versus the desired 455 kHz signal. This can be a challenging signal to use as a trigger. Using the 150 kHz highpass filter, the 50 or 60 Hz component is attenuated by several orders of magnitude, greatly improving the trigger stability and jitter performance of the acquisition.

HF Reject Example

An example application for the HF Reject is triggering on a 1 kHz Sigma-Delta integrator output in the presence of high-frequency signal content, including overshoot and ringing on the transitions. When this high frequency content crosses over the trigger threshold, false triggers occur. Using the 150 kHz lowpass filter, you can reject the high-frequency content, noise, and overshoot to obtain a clean trigger source.

NI 5160/5162 Trigger Holdoff

For the NI 5160/5162, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition. It is meaningless to apply a trigger holdoff on a single-record or continuous acquisition.

The NI 5160/5162 has a maximum trigger holdoff of 27.4 s and a minimum trigger holdoff of 6.4 ns.

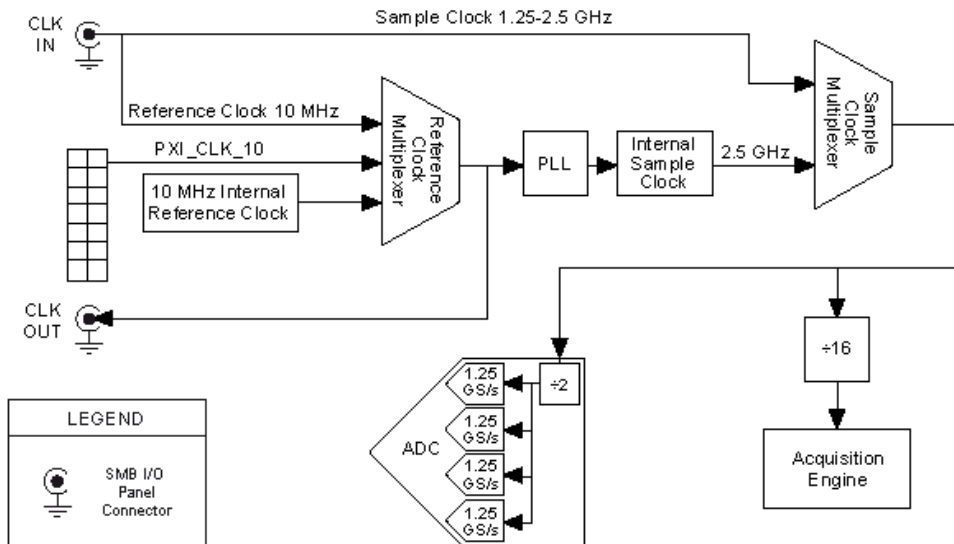
NI 5160/5162 Trigger Delay

[Trigger delay](#), which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify.

The maximum trigger delay is 73,786,976 seconds (28 months).

NI 5160/5162 Clocking

The clock circuitry on the NI 5160/5162 offers versatile clocking options with the ability to use either the internal or external sources for sample clocks and reference clocks. The following diagram shows the clocking options of the NI 5160/5162.



Sample Clock

The sample clock is sent to the ADC, where it is divided by 2 and distributed to the 4 ADC quadrants. The sample clock is divided by 16 and sent to the digitizer [acquisition engine](#). The NI 5160/5162 can decimate sample clocks (internal or external) by an integer divisor.

Internal Sample Clock

The internal 2.5 GHz sample clock is [phase locked](#) to the onboard 10 MHz oscillator by default. Alternatively, the sample clock may be phase locked with the PXI Express 10 MHz clock or an external reference clock provided through the SMB CLK IN connector on the front panel of the device. The internal sample clock is decimated using the [Min Sample Rate](#) property.

External Sample Clock

Some applications may require sampling at specific sample rates that cannot be achieved by decimating the internal sample clock by an integer divisor. In these cases the digitizer can accept an external sample clock. External clocking also provides another method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel

SMB CLK IN connector. Refer to the hardware specifications document for external sample clock requirements.



Note Exporting the external sample clock is not supported on the NI 5160/5162.

You can specify an external sample clock for use by configuring the [Sample Clock Timebase Source](#) and [Sample Clock Timebase Rate](#) properties. An external sample clock is decimated or multiplied by the values of the [Sample Clock Timebase Divisor](#) and [Sample Clock Timebase Multiplier](#) properties, respectively.

Reference Clock

The [reference clock](#) is used in the digitizer's phase-locked loop (PLL) circuit to lock the sample clock to the reference clock. The frequency stability of the sample clock matches that of the reference clock when the two are phase locked. This digitizer can accept a reference clock from the front panel (CLK IN) as well as from PXI_CLK10. A reference clock provided to the CLK IN SMB connector is typically 10 MHz. The PXI_CLK10 signal is always a 10 MHz clock. Phase locking can lock sample clocks of multiple devices to the same reference clock. For PCI devices, the onboard 10 MHz oscillator is the default reference clock. For PXI and PXIe devices, the default reference clock behavior varies by device.



Note Locking to a reference clock is not valid when using external sample clocking.

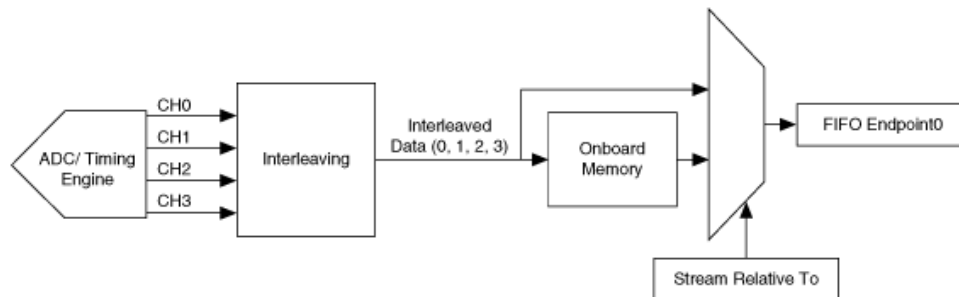
Exporting Reference Clock

You can export the reference clock to synchronize other instruments to the NI 5160/5162. For more information on exporting the reference clock, refer to the [NI 5160/5162 Routing Matrix](#).

NI 5160/5162 Peer-to-Peer Streaming

Peer-to-peer (P2P) streaming technology uses PCI Express to enable direct, point-to-point transfers between multiple instruments without sending data through the host processor or memory. This enables devices in a system to share information without burdening other system resources.

The NI 5160/5162 supports peer-to-peer streaming using the NI-P2P API. For more information about using NI-P2P to stream data between devices, refer to the **Peer-to-Peer Streaming Help**. The **NI-SCOPE Function Reference Help** and the **NI-SCOPE LabVIEW Reference Help** provide more information on configuring the digitizer's endpoint resources to send data to another peer.



NI 5160/5162 Configuring an Endpoint

To set properties on an endpoint, you must select that endpoint using the [Active Channel](#) property and the endpoint name `FIFOEndpoint0`.

Considerations When Configuring an Endpoint

Keep the following points in mind when configuring an endpoint:

- Set the [P2P Enabled](#) property or the `NISCOPE_ATTR_P2P_ENABLED` attribute to TRUE to activate the P2P endpoint and to use any of the P2P features.
- Use the [Channels to Stream](#) property or the `NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM` to determine the specific data that is sent to an endpoint.
- Use the [Binary Sample Width](#) property or the `NISCOPE_ATTR_BINARY_SAMPLE_WIDTH` attribute to set the sample width.
- Call `niScope Abort`, `niScope Close`, or `niScope Reset` on the digitizer session to reset an endpoint.

On the NI 5160/62 all enabled channels are streamed over `FIFOEndpoint0` when P2P is enabled. [Channels to Stream](#) must be identical to the enabled channels, which is the default value. Channels are interleaved in ascending order.

Configuring a Peer-to-Peer Stream

To configure a peer-to-peer stream using the NI-P2P API, a writer and reader handle for each endpoint are required. Use the [niScope Get Stream Endpoint Handle VI](#) to get a writer endpoint handle. Refer to the **Peer-to-Peer Streaming Help** for more information on creating and monitoring peer-to-peer streams.

Configuring a Trigger with a P2P Stream

NI 5160/5162 supports the following modes of P2P which you can select using the [Stream Relative To](#) property:

- Start Trigger
- Reference Trigger
- Sync Trigger



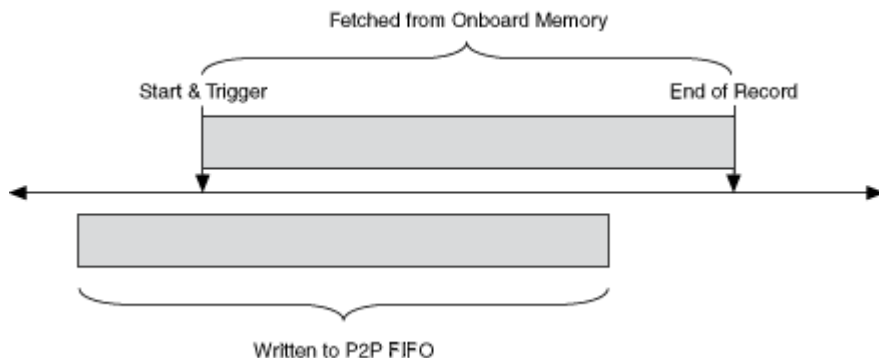
Note During a TClk acquisition, the time between the sync and reference triggers can vary by up to 1 TClk period. Data between devices will be aligned, but the reference trigger position can vary by up to 1 TClk period between records.

Start Trigger

In Start Trigger mode, all data is streamed from the start trigger until the end of post-reference trigger sampling. This mode is same as the P2P streaming behavior of the NI 5122/5622. Depending on the type of trigger configured for the acquisition, data is not written to the P2P endpoints in the same way as it is written to onboard memory.

Immediate Trigger

Because of differences in the data path leading to the P2P endpoints and the onboard memory, the first sample written to the P2P endpoints is from an earlier time than the first sample written to onboard memory.

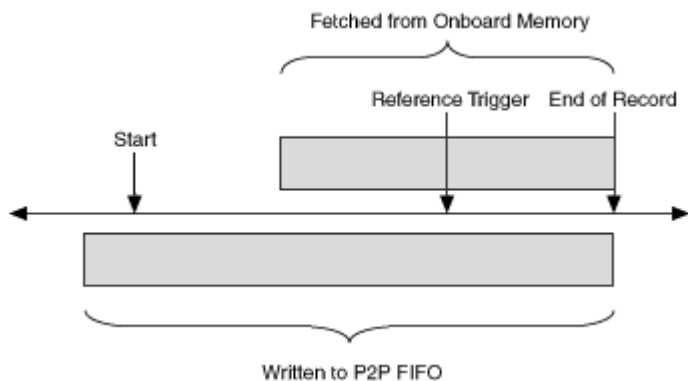


Note The record sent to the onboard memory is the same length as the record sent to the P2P FIFO.

All data written to the P2P endpoints is valid, but the samples are not aligned with the same record stored in onboard memory.

Configured Reference Trigger

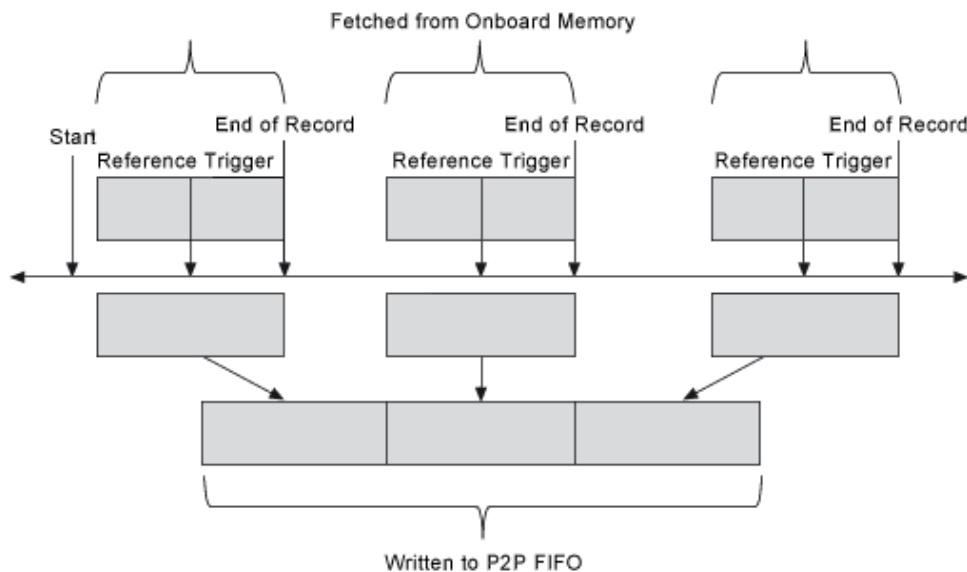
While streaming relative to Start, when the trigger type is anything other than immediate, the digitizer sends all data between the Start trigger and the End of Record event to the endpoint.



Reference Trigger

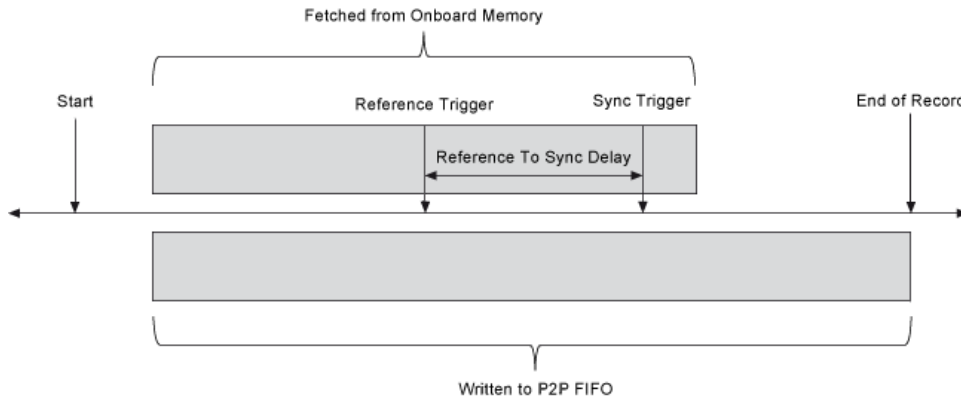
In Reference Trigger mode, only pre-reference trigger and post-reference trigger samples are streamed relative to the reference trigger. In this mode, data that would be fetched from memory is streamed over P2P. The amount of data streamed per record can be determined by reading the [P2P Samples Transferred Per Record](#) attribute.

In a multi-record acquisition, the NI 5160/5162 sends data records continuously to the peer. For example, if a user requests three records, 1000 samples, and a 50% reference position, the NI 5160/5162 will transfer 3000 samples. Each record will be 1000 samples long, with 500 pre-reference trigger and 500 post-reference trigger samples. In this example, record 0 is samples 1 to 1000, record 1 is samples 1001 to 2000, and record 3 is samples 2001 to 3000.



Sync Trigger

In Sync Trigger mode, pre-reference trigger and post-reference trigger samples are streamed relative to the sync trigger, but extra pre-reference trigger samples are added to account for the variability in trigger position with TClk. You must use the Sync Trigger mode with TClk to synchronize data across multiple boards. The amount of data streamed per record can be determined by reading the [P2P Samples Transferred Per Record](#) attribute. The “Reference to Sync Delay” time value can vary between 1 and 2 TClk periods which means that 2 TClk periods of pre-reference trigger samples are added to every streamed record to prevent missing pre-reference trigger samples.



Scaling and Normalization of Binary Data

In a normal acquisition, you cannot assume that the binary data maps perfectly to the vertical range. For example, if you fetch 16-bit binary data, you might expect that the positive full scale binary value corresponds to the maximum positive voltage of the vertical range, and that the negative full scale binary value corresponds to the maximum negative voltage. This is not the case, however, for a number of reasons.

First, the full scale of the ADC is not mapped to the vertical range. A few codes on the positive and negative ends are left as headroom, in case the input signal slightly exceeds the specified vertical range. Second, the binary values do not account for the required corrections to gain and offset based on calibration data. If you fetch binary data, you can convert to voltage values using the gain and offset values from the `wfmInfo` struct or use the [niScope Get Scaling Coefficients VI](#) or the [niScope_GetScalingCoefficients](#) function to get these coefficients. These values account for the ADC normalization and calibration data.

$$\text{Voltage} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

To get binary data that accounts for both ADC normalization and calibration, use the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function. Applying gain and offset from the normalization coefficients results in a range of binary values that maps exactly to the vertical range you configure.

$$\text{Normalized Binary Data} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

$$\text{Voltage} = \text{Normalized Binary Data} \times (\text{Range} / 2^{\text{resolution in bits}})$$

In a normal acquisition, you must know the gain and offset values to correctly convert the binary data to voltage values. After applying gain and offset values from the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function, you can assume the following:

- The maximum positive binary value maps to the maximum positive voltage of the vertical range.
- The maximum negative binary value maps to the maximum negative voltage of the vertical range.
- The vertical range is divided evenly across the possible binary values.

Peer-to-Peer Streaming VIs/Functions

LabVIEW	C/C++
Streaming Session—Use the following NI-P2P VI to configure a peer-to-peer streaming session.	
niScope Get Stream Endpoint Handle VI	niScope_GetStreamEndpointHandle
Scaling and Normalization—Data written to a peer-to-peer endpoint is not scaled or calibrated. Use the following VIs/functions to scale or normalize the binary data.	
niScope Get Wfm Coefficients (poly) VI	niScope_GetNormalizationCoefficients
	niScope_GetScalingCoefficients

Peer-to-Peer Streaming Properties/Attributes

Use the following properties and attributes to configure a peer-to-peer streaming session.

LabVIEW	C/C++
P2P Enabled	NISCOPE_ATTR_P2P_ENABLED
Channels to Stream	NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM
Endpoint Size	NISCOPE_ATTR_P2P_ENDPOINT_SIZE
Samples Available In Endpoint	NISCOPE_ATTR_P2P_SAMPLES_AVAILABLE_IN_ENDPOINT
Most Samples Available In Endpoint	NISCOPE_ATTR_P2P_MOST_SAMPLES_AVAILABLE_IN_ENDPOINT

<u>Samples Transferred</u>	<u>NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED</u>
<u>Samples Transferred Per Record</u>	<u>NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED_PER_RECORD</u>
<u>Endpoint Overflow</u>	<u>NISCOPE_ATTR_P2P_ENDPOINT_OVERFLOW</u>
<u>FIFO Endpoint Count</u>	<u>NISCOPE_ATTR_P2P_FIFO_ENDPOINT_COUNT</u>
<u>Stream Relative To</u>	<u>NISCOPE_ATTR_P2P_STREAM_RELATIVE_TO</u>
<p>Manual—The following property/attribute activates advanced functionality for a P2P endpoint. To use any of the P2P advanced features, this property/attribute must be set to TRUE. This attribute cannot be used for an endpoint if that endpoint is configured using NI-P2P.</p>	
<u>Manual Configuration Enabled</u>	<u>NISCOPE_ATTR_P2P_MANUAL_CONFIGURATION_ENABLED</u>
<p>Configuration—The following properties/attributes provide a way to configure a destination for endpoint data without using NI-P2P. These attributes can be used to send data to a DMA window on a third-party device, or to a memory window on the host machine.</p>	
<u>Data Transfer Permission Address</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR</u>
<u>Data Transfer Permission Address Type</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR_TYPE</u>
<u>Destination Window Address</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR</u>
<u>Destination Window Address Type</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_TYPE</u>
<u>Destination Window Size</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_SIZE</u>
<p>Notification—The following properties/attributes can be used to send a message corresponding to an event directly to a peer or to a location in host memory.</p>	
<u>Push Message On</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON</u>
<u>Message Push Address</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR</u>
<u>Message Push Address Type</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDRESS_TYPE</u>

Message Push ValueNISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VALUE

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

Making Multiple-Record Acquisitions

NI 5160/5162 Onboard Memory

The NI 5160/5162 allocates at least 512 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is approximately 256 16-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 256 samples. Refer to the [hardware specifications document](#) for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. These samples are known as the "pre-reference trigger" samples. After the trigger is received, the NI 5160/5162 continues to acquire post-reference trigger samples. The relationship of the number of pre-reference trigger samples to post-reference trigger samples is specified by the "Reference position" attribute (by default, 50% of the record length is post-trigger.) The acquired samples are placed into onboard memory. The number of pre-reference trigger or post-reference trigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

Features Supported by SMC-Based Devices and USB Devices

NI 5160/5162 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes exceed the temperature range specified or time exceeds the time period specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 $^\circ\text{C}$ range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 $^\circ\text{C}$ range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 $^\circ\text{C}$, assuming the Tempco (TC) error is specified as

$$\text{TC} = (0.1\% \text{ of input} + 1 \text{ mV}) / \text{ }^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / \text{ }^\circ\text{C)}$$

The additional error is

$$20 \text{ }^\circ\text{C} \times \text{TC} = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 $^\circ\text{C}$ ambient temperature.



Note The specifications listed above are only used to illustrate an example. For device-specific gain accuracy and offset drift specifications, refer to the [device specifications](#).

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5160/5162, self-calibration yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This is done for all ranges, both input impedance paths (50 Ω and 1 M Ω), and all filter paths (enabled/disabled).
- Calibrates trigger level offset and gain.
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Calibrates gain, offset, and phase for interleaved ADC modes, reducing ADC interleaving errors in [TIS](#) modes.
- Calibrates the phase DAC used to adjust the phase of the sample clock.
- Takes approximately 2 minutes to complete.

When to Self-Calibrate

For optimum performance, use self-calibration in the following circumstances:

- When the digitizer is placed in a new system and has warmed up for at least 15 minutes
- Any time the temperature changes more than 3 $^{\circ}\text{C}$ from the previous self-calibration (refer to the [device specifications](#) for specific temperature ranges, as some specifications may have different temperature limits)
- 90 days after the previous self-calibration

The result is a product that yields full performance over its operating temperature range and two-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required to ensure performance that is within specification over the next year.

The NI 5160/5162 has a temperature sensor that monitors temperature variations and can be read through the NI-SCOPE "Device Temperature" attribute. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

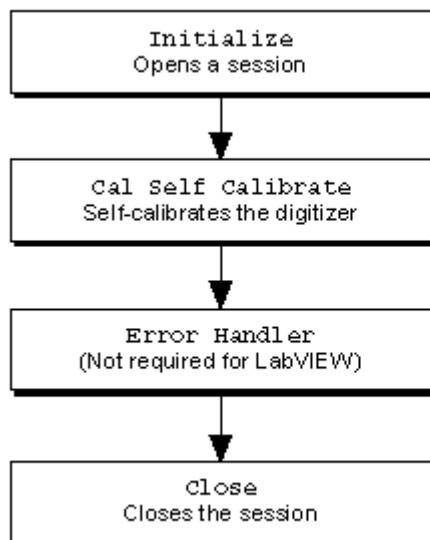
Input Connections During Self-Calibration

The NI 5160/5162 internal circuitry is automatically isolated from the input during self-calibration. However, if high-voltage, high-frequency signals are present during self-calibration, the calibration results may be adversely affected or the calibration may fail with an error.

When possible, always disconnect the inputs.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference. NI 5160/5162 (2 CH)—Calibrates the external analog trigger channel.	Once every two years	Calibrates and verifies to full specifications
Self-calibration	Offset and gain Trigger level Trigger timing Interleaved ADC gain, offset, and phase Sample clock phase ADC	90 days, or when temperature changes >3 °C	Ensures range to range matching Ensures trigger accuracy
No calibration	None, within 2-year calibration cycle or if temperature stays within ± 3 °C	High absolute accuracy not required outside of 3 °C	If self-calibration is not used, derate the absolute accuracy using the specified temperature coefficient (Tempco)

NI 5185/5186

The NI PXIe-5185 and NI PXIe-5186 were developed jointly between Tektronix and National Instruments. The devices use Tektronix, Enabling Technology™ to deliver wide analog bandwidth and high-speed sampling on the National Instruments Synchronization and Memory Core (SMC) technology with TClk synchronization.

Features

The NI 5185/5186 has the following features:

- 2 input channels
- 8-bit vertical resolution
- 12.5 GS/s (one channel) or 6.25 GS/s (two channels) sample rate

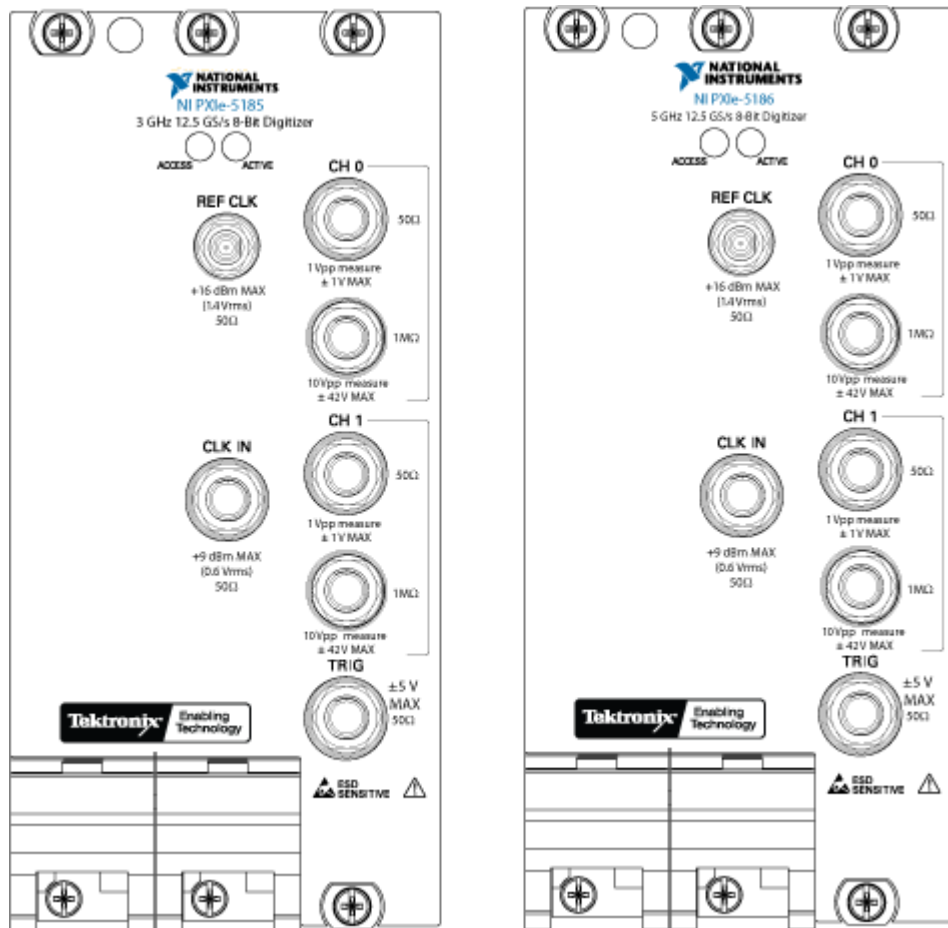
- x4 PXI Express
- 32 MB or 1 GB onboard memory
- Bandwidth
 - NI 5185: 50 Ω at 3 GHz
1 M Ω at 500 MHz
 - NI 5186: 50 Ω at 5 GHz
1 M Ω at 500 MHz
- Analog edge and external trigger with sub-sample clock accuracy

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5185/5186 Front Panels

The following figures show the front panels of the NI 5185/5186. Descriptions of the LEDs and connectors are shown below.



Previous NI 5185/5186 Modules

Early versions of the NI 5185/5186 only support 50Ω input signals. If the front panel of your NI 5185/5186 looks different from the front panels shown above, you may have an earlier version of the device. Check the part number of your device to be sure:

- NI 5185 module part numbers 199363x-0zL and NI 5186 module part numbers 193537x-0zL (where **x** is any letter and **z** is any number) only support 50Ω input signals.
- NI 5185 module part numbers 152962x-0zL and NI 5186 module part numbers 152961x-0zL (where **x** is any letter and **z** is any number) support both 50Ω and 1 MΩ input signals.

LEDs

The NI 5185/5186 has two LEDs to indicate status: Access and Active.

Access LED

The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The device is being accessed or initialized. The Access LED is solid amber during initialization (for approximately 5–10 minutes after startup) and becomes green when this initialization period is complete. While the LED is solid amber, any host-initiated access to daughtercard registers will cause the LED to turn off for about 50 ms. If the steady state was green (that is, initialization was complete), then the LED flashes amber for 50 ms.
Green	The module is ready to be programmed by NI-SCOPE.

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Reference (Stop) trigger.
Green	The module has received a Reference (Stop) Trigger, so it is acquiring a waveform.
Red	The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:

- Device initialization in progress—during start-up, the Active LED will appear red for a short period of time.
- PLL unlocked—the module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error.
- External sample clock error—the module is unable to detect the external sample clock.
- Overheating error—the module has detected an overheating error. Refer to [Thermal Shutdown](#) for more information.



Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.

Connectors

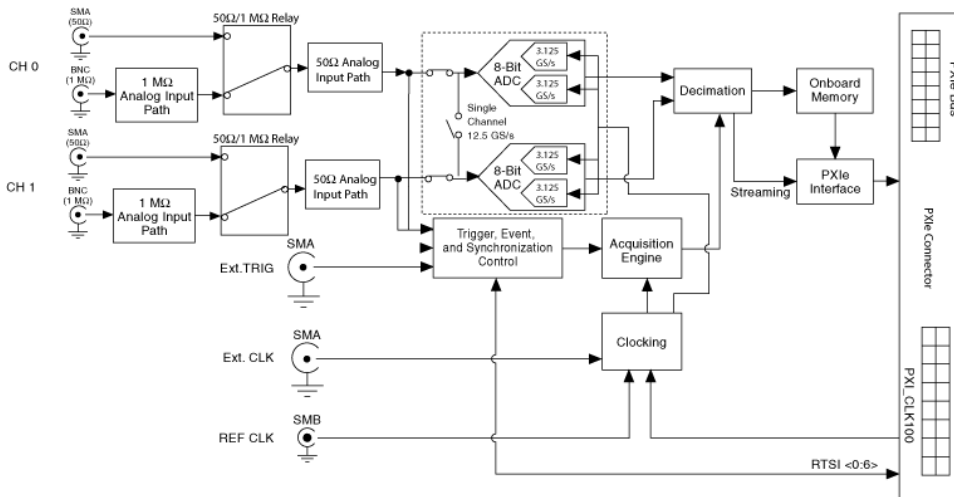
The NI 5185/5186 has the following connectors on the front panel.

Connector	Description	Function
REF CLK	SMB jack	Imports an external reference clock to the digitizer.
CLK IN	SMA female	Imports an external sample clock to the digitizer.
CH 0	SMA female (50Ω)	Analog input connection; digitizes data and triggers acquisitions.
CH 0	BNC (1 MΩ)	Analog input connection; digitizes data and triggers acquisitions.
CH 1	SMA female (50Ω)	Analog input connection; digitizes data and triggers acquisitions.

CH 1	BNC (1 M Ω)	Analog input connection; digitizes data and triggers acquisitions.
TRIG	SMA female	External analog trigger connection.

NI 5185/5186 Block Diagram

The following figure shows a detailed block diagram of the NI 5185/5186.



Some previous NI 5185/5186 devices do not support 1 M Ω inputs. For more information, refer to [NI 5185/5186 Front Panels](#).

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

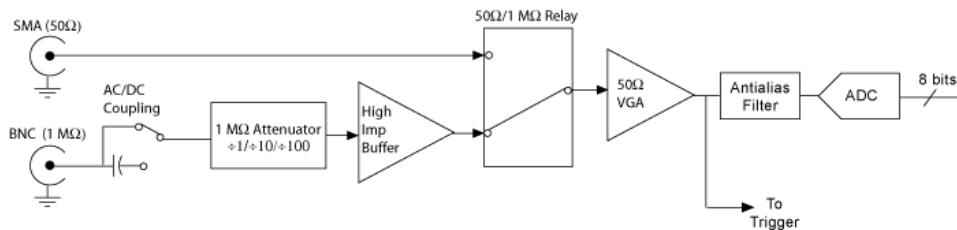
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5185/5186 Input Signal Conditioning

The NI 5185/5186 provides the following input channel signal conditioning path.



Some previous NI 5185/5186 devices do not support 1 M Ω inputs. For more information, refer to [NI 5185/5186 Front Panels](#).



Note The ground on the device inputs is connected to the chassis ground. For information regarding safety considerations when grounding, such as maximum voltage between the device inputs and the device ground, refer to the "Safety" section of the device specifications.

NI 5185/5186 Input Ranges

The NI 5185/5186 supports the following input ranges:

- 50 Ω path: 110 mV_{pp} to 1 V_{pp}
- 1 M Ω path: 110 mV_{pp} to 10 V_{pp}

Some previous NI 5185/5186 devices do not support 1 M Ω inputs. For more information, refer to [NI 5185/5186 Front Panels](#).

NI 5185/5186 Input Impedance

The NI 5185/5186 has 50 Ω and 1 M Ω input impedances.

Some previous NI 5185/5186 devices do not support 1 M Ω inputs. For more information, refer to [NI 5185/5186 Front Panels](#).

NI 5185/5186 Coupling

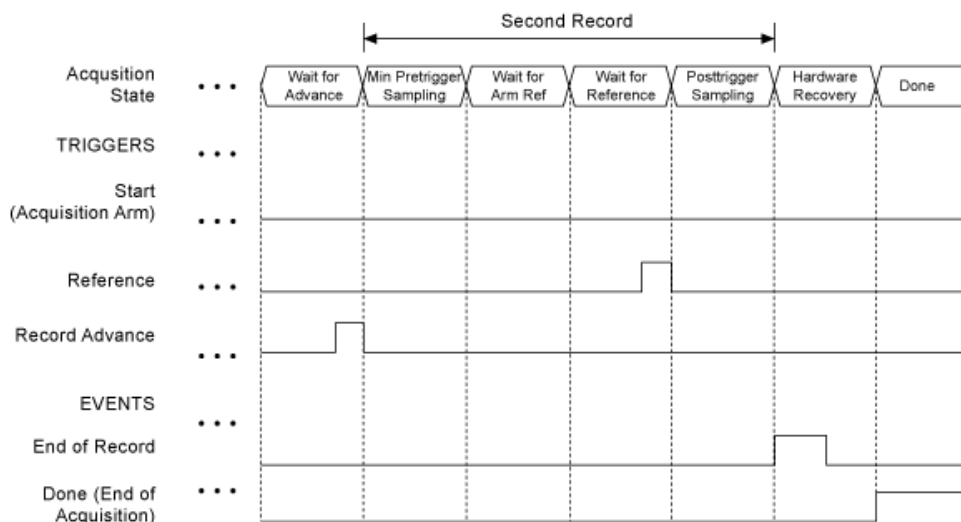
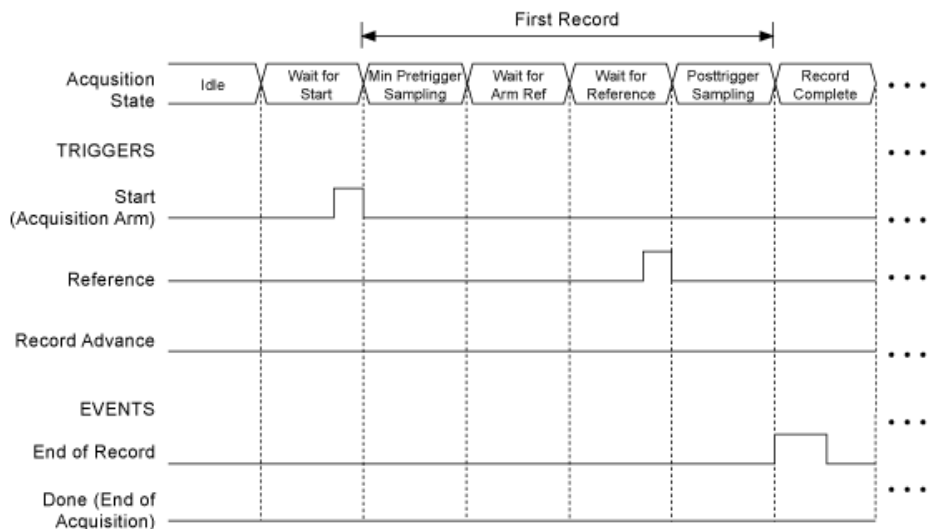
The NI 5185/5186 has DC [input coupling](#) for the 50 Ω input path and both AC and DC coupling for the 1 M Ω path.

Some previous NI 5185/5186 devices do not support 1 M Ω inputs. For more information, refer to [NI 5185/5186 Front Panels](#).


NI 5185/5186 Timing Diagram

The NI 5185/5186 supports multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how the NI 5185/5186 reacts to the user-configurable input triggers during a multirecord acquisition.




Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. If an input trigger is not explicitly configured, the device will default to transition immediately into the next state. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#). The digitizer is only able to detect a particular trigger when in that trigger's appropriate **Wait for** state. For example, the digitizer is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state. Triggers in states that are not prefixed by **Wait for** are generated by the digitizer, and can be exported.

 **Note** The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI 5185/5186 Routing Matrix

The following table shows the signals available for export from the NI 5185/5186.

Source	Destination
	PXI_Trig <0..6> (PXI Bus)
Triggers	
Acquisition Arm (Start) Trigger	√
Reference (Stop) Trigger	√
Events	
End of Record Event	√
End of Acquisition Event	√
Ready for Start Event	√
Ready for Reference Event	√
Ready for Advance Event	√
 Note The NI 5185/5186 can only export/import 4 events/triggers simultaneously from/to the PXI bus.	

NI 5185/5186 Trigger Types

The NI 5185/5186 supports analog [edge](#), [immediate](#), [software](#), and [digital](#) triggers.

End of Record to Advance Trigger Holdoff

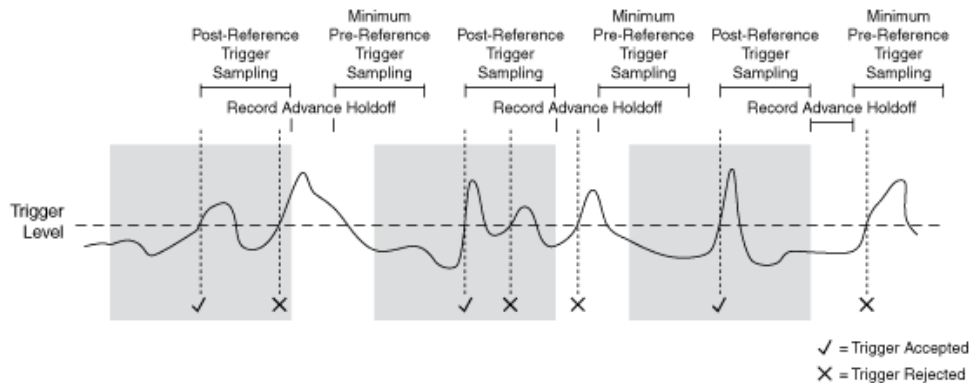
End of Record to Advance Trigger Holdoff is the length of time (in seconds) that a device waits between the completion of one record and the acquisition of pre-trigger samples for the next record. During this time, the acquisition engine state delays the transition to the Wait for Advance Trigger state, and will not store samples in onboard memory, accept an Advance Trigger, or trigger on the input

signal. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

For the NI 5185/5186, you cannot fetch data from a device before an acquisition completes if the sample rate is greater than 2GS/s with 1 channel enabled or 1GS/s with 2 channels enabled. The device cannot maintain simultaneous reads and writes to onboard memory above those sample rates. However, fetching during acquisition above these sample rates can be achieved by using End of Record to Advance Trigger Holdoff.

Only the NI 5185/5186 supports record End of Record to Advance Holdoff. You can use End of Record to Advance Holdoff to fetch data from the device before a multi-record acquisition is complete. When End of Record to Advance Holdoff is used, the device can transfer data during the holdoff time between records.

Assuming the End of Record to Advance Holdoff time is greater than dead time, refer to the following figure for acquisition triggering behavior.



Use the following equation to calculate minimum time between triggers.

Minimum Time Between Triggers = Time Per Record + End of Record to Advance Holdoff

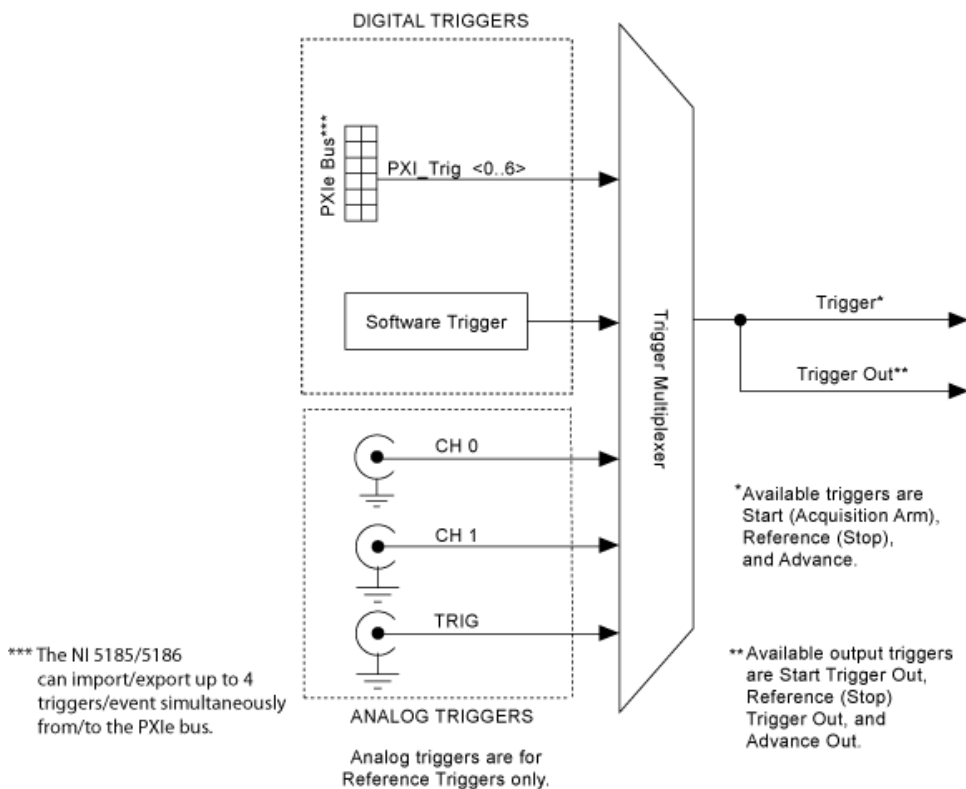
Related Topics

[Making Multiple-Record Acquisitions](#)

[SMC-Based Digitizers Acquisition Engine State Diagram](#)

NI 5185/5186 Trigger Sources

The following figure shows the trigger sources for the NI 5185/5186.



NI 5185/5186 Trigger Holdoff

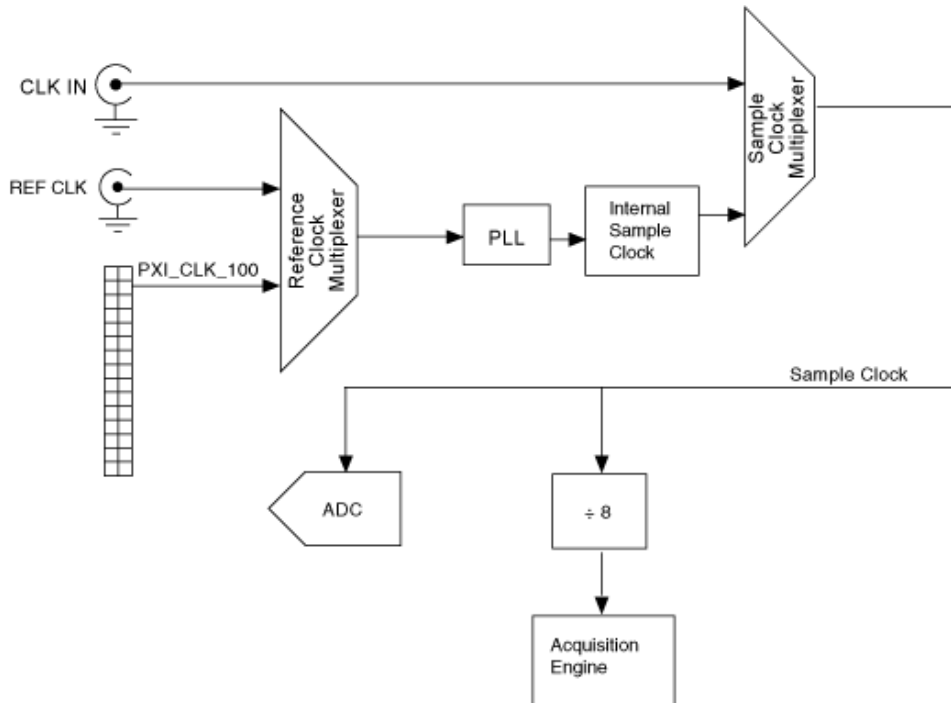
The NI 5185/5186 has a maximum trigger holdoff of 10.99 s.

NI 5185/5186 Trigger Delay

Trigger delay, which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. The maximum trigger delay for the NI 5185/5186 is 1,300,000 seconds.

NI 5185/5186 Clocking

The following figure shows the clock circuitry on the NI 5185/5186.



Sample Clock

The sample clock is sent to the ADC of each channel and to the input timing engine. The NI 5185/5186 can decimate its sample clock (internal or external) by an integer divisor. When using an external clock, you can decimate to achieve rates below the external clock frequency. You can also achieve a higher effective sample rate by multiplying the sample clock by 1, 2, or 4 (in one-channel mode), which interleaves the ADCs.

Internal Sample Clock

NI 5185/5186 has an onboard VCO running at a default frequency of 3.125 GHz. By default this VCO is phase locked to PXI_CLK100 through a PLL circuit. The internal sample clock can also be phase locked to a user provided external reference clock. Locking to an external reference clock is useful when synchronizing the digitizer with other devices in a measurement system, or if frequency accuracy greater than PXI_CLK100 is required.

External Sample Clock

Some applications may require sampling from an external timebase. In these cases the NI 5185/5186 can accept an external sample clock. External clocking provides a method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. Depending on the quality of the clock source, use of an external sample clock can also improve phase noise and spectral performance. An external clock can be supplied to the digitizer at the front panel connector (CLK IN). Refer to the specifications document that shipped with your device for external sample clock requirements.

Reference Clock

The reference clock is used in the NI 5185/5186 phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. By default the PLL locks to PXI_CLK100. The NI 5185/5186 can accept a reference clock from its front panel (REF CLK) as well. Refer to the specifications document that shipped with the device for information about the reference clock frequency. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. Phase locking multiple devices to the same reference clock synchronizes those devices.

Exporting Reference Clock

The NI 5185/5186 cannot export its reference clock.

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger

of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5185/5186 Calibration

On every power cycle, the NI PXIe-5185/5186 performs an initialization routine that optimizes the spectral performance of the digitizer. This initialization routine may impact the accuracy of the DC gain and offset calibration constants stored in the device from the previous power cycle. Therefore, it is recommended that a self-calibration be performed after each power cycle after the 25 minute warm-up time has been met. This self-calibration will ensure that the digitizer achieves its optimum DC accuracy performance and meets the specifications detailed in the published specifications document.

External calibration compares and calibrates the internal references on the NI PXIe-5185/5186 to an external, very accurate reference. Self-calibration uses these internal references to correct for DC and gain errors during its calibration routine. External calibration therefore ensures that the results obtained from self-calibration are accurate over the recommended external calibration time period. External calibration also calibrates the external trigger input levels and the overload protection limits for the analog input channels. It is recommended that external calibration be performed annually.

The NI PXIe-5185/5186 performs within its specifications over a finite temperature range and time period. If the temperature changes and time exceeds those specified, and your application requires tight specifications, calibration is required.

For example, the accuracy of the NI PXIe-5186 is specified as

$\pm(2\% \text{ of input} + 0.35\% \text{ FS} + 0.7 \text{ mV})$ within $\pm 3 \text{ }^\circ\text{C}$ of self-calibration.

This means that if you apply 200 mV to the input using the 1 V_{pk-pk} range, the error is:

$2\% \text{ of } 200 \text{ mV} + 0.35\% \text{ of } 1 \text{ V} + 0.7 \text{ mV} = \pm 8.2 \text{ mV}$ for a temperature range within $\pm 3 \text{ }^\circ\text{C}$ of self-calibration.

When a system is composed of multiple integrated instruments the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors. Therefore, it is possible that the digitizer may be exposed to an ambient temperature that is outside $\pm 3 \text{ }^\circ\text{C}$ of the last self calibration. If the ambient temperature is outside of the

± 3 °C range of the last self calibration, you can use the temperature coefficient of the digitizer to know exactly what the measurement accuracy is to compensate for this temperature variation. The temperature coefficient of the NI PXIe-5185/5186 is specified as

$$TC = \pm(0.23\% \text{ of input} + 0.03\% \text{ of FS}) / ^\circ\text{C}$$

In the example above, if the ambient temperature of the digitizer is 48 °C and the last self-calibration was done at 28 °C the additional error due to temperature variation is

$$20\text{ }^\circ\text{C} \times TC = 20 \times \pm(0.23\% \text{ of } 200\text{ mV} + 0.03\% \text{ of } 1\text{ V}) = \pm 15.2\text{ mV}$$

The total error is almost three times the specified error (± 23.4 mV in the example above, versus ± 8.2 mV if temperature effect is ignored) due to the 48 °C ambient temperature. If the additional error due to temperature drift is unacceptable for your application, you can perform a self-calibration at the new temperature and get your accuracy back to within the specified ± 8.2 mV.

The temperature of the last self-calibration and external calibration, as well as the current device temperature, can be found in the Calibration tab in Measurement & Automation Explorer (MAX). The calibration temperatures can also be read programmatically through LabVIEW using the niScope Cal Fetch Temperature VI. The current temperature of the device can be read through LabVIEW by selecting the device temperature property in the NI-SCOPE property node.

When to Self-Calibrate

For optimum performance, use self-calibration in the following circumstances:

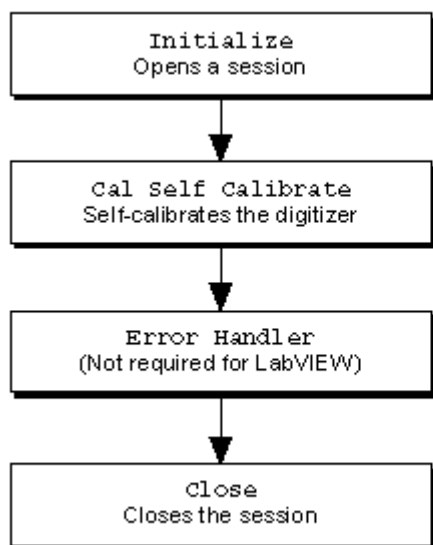
- Every time the system is powered on and the module has warmed up at least 25 minutes.
- When the digitizer is placed in a new system.
- When the temperature changes more than 3 °C from the previous self-calibration
- 90 days after the previous self-calibration.

The result is a product that yields full performance over its operating temperature range and one-year calibration cycle for DC accuracy, AC response, and trigger level/

timing. When the one-year calibration interval expires, an external calibration is required.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available for the NI 5185/5186 and when to use them is shown in the following table.

Calibration	Impact	When
External calibration	Calibrates the external trigger, the onboard references that are used in self-calibration, and the input overload levels.	Once a year
Self-calibration	Calibrates DC Gain, DC Offset, Internal Trigger Timing, and ADC alignment spurs if temperature	<ul style="list-style-type: none"> After a 25 minute warmup period from each power cycle

	e stays within ± 3 °C and the module is not power cycled	<ul style="list-style-type: none"> ▪ Every 90 days ▪ When temperature changes more than 3 °C
No calibration	None, within 1 year calibration interval and if temperature stays within ± 3 °C and the module is not power cycled	High accuracy not required outside of 3 °C between power cycles

NI 5622

NI 5622 high-speed digitizers with onboard signal processing feature large dynamic range, wide analog bandwidth, and deep onboard acquisition memory for frequency domain measurements. The NI 5622 can be used in conjunction with the NI 5601 downconverter (as the NI PXI-5663 RF signal analyzer) or as a standalone instrument.

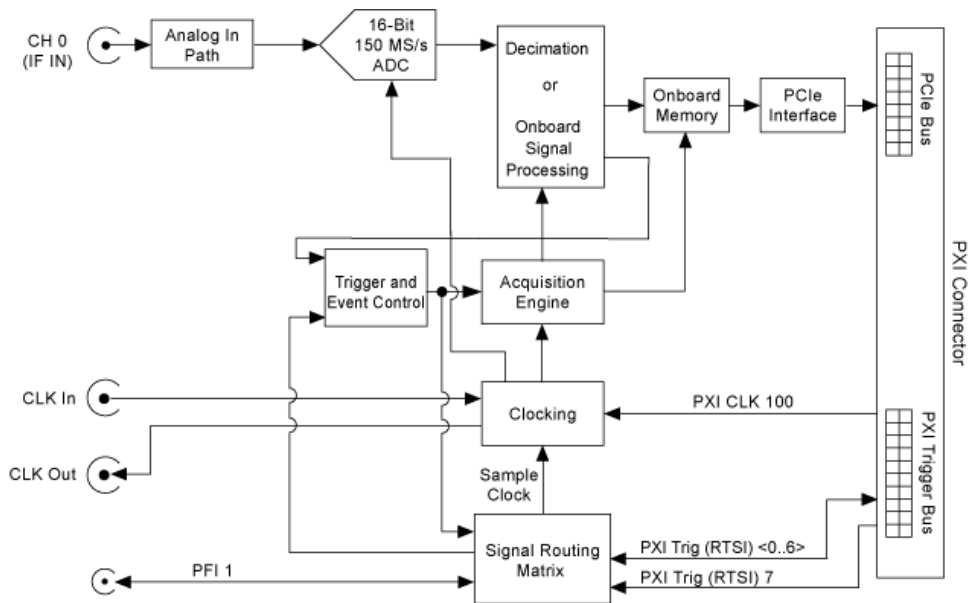
Features

The NI 5622 has the following features:

- 1 channel
- 16-bit vertical resolution
- 150 MS/s real-time sampling rate
- 3-250 MHz direct path frequency range or 50 MHz alias-protected IF bandwidth, centered at 187 MHz
- 64 or 256 MB of memory per channel
- PXI Express
- [Onboard signal processing](#), including digital downconversion and fractional resampling
- [NI-TClk synchronization](#)

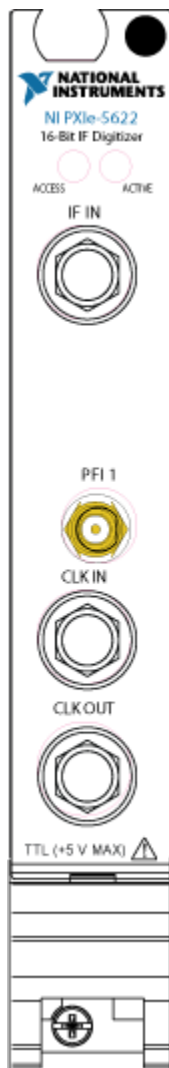
NI PXIe-5622 Block Diagram

The following figure shows a detailed block diagram of the NI PXIe-5622.



NI PXIe-5622 Front Panel

The following figure shows the front panel of the NI PXIe-5622. Descriptions of the LEDs and connectors are shown below.



LEDs

The NI 5622 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked PLL. A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—the module is unable to detect the external sample clock. ▪ Overheating error—the module has detected an overheating error. Refer to Thermal Shutdown for more information. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>

Connectors

The NI 5622 has the following connectors on the front panel.

Connector	Description	Function
IF IN	SMA connector	Analog input connection; digitizes data and triggers acquisitions
PFI 1	SMB connector	Digital trigger connection
CLK IN	SMA connector	Imports an external reference or sample clock to the digitizer
CLK OUT	SMA connector	Exports the digitizer reference or sample clock

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.
- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

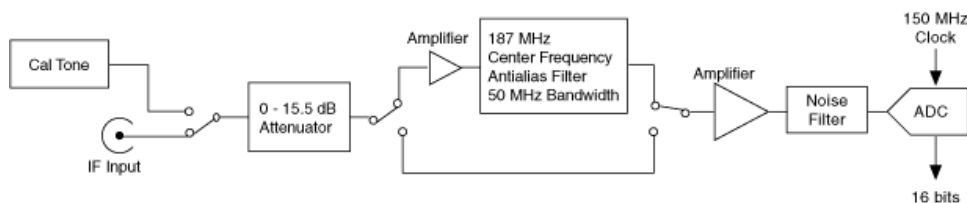
- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5622 Signal Conditioning

The NI 5622 provides two input signal conditioning paths, one filtered and one unfiltered.



The IF filtered path, shown in the upper signal conditioning path, has a bandwidth of 50 MHz centered at 187 MHz. This path can be used with NI 5663 vector signal analyzer applications, or other applications requiring alias-protected sampling in the third Nyquist zone for the 150 MS/s ADC. The third Nyquist zone is 150 MHz to 225 MHz. To enable the IF filtered path, set the Bandpass Filter Enabled property to TRUE.

The direct path, shown in the lower signal conditioning path, has a frequency range of 3 to 250 MHz. This path can be used in digitizer-only applications. When the direct path (unfiltered) is selected, you should perform antialias filtering, external to the digitizer, depending on the spectral content of the signal being digitized.



Note The ground on the device inputs is connected to the chassis ground.

Related Information

[Bandpass Filter Enabled Property](#)

NI 5622 Input Range

The NI PXIe-5622 has three input ranges: 1 dBm, 4 dBm, and 7 dBm. These ranges are equivalent to $0.71 V_{pk-pk}$, $1 V_{pk-pk}$, and $1.4 V_{pk-pk}$, respectively.

NI 5622 AC/GND Coupling

You can select AC or GND [input coupling](#) for the 50 Ω input path.

Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

NI 5622 Input Impedance

The NI PXIe-5622 has a 50 Ω input impedance.

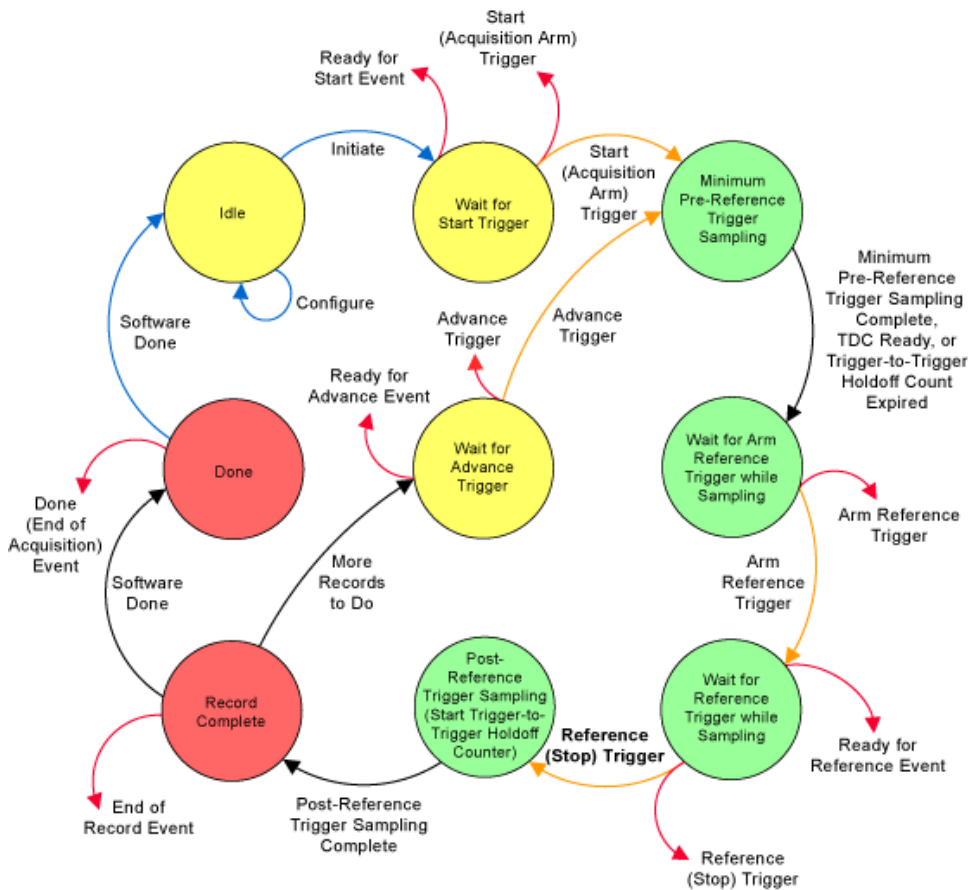
Protection

The 50 Ω input of the NI 5622 is protected against overvoltage conditions. Refer to the **NI 5622 Specifications** for the specific limits. If an overvoltage event is large and sudden enough, however, the protection circuit might not prevent permanent

damage to the hardware. It is therefore important that you observe the specified maximum signal input levels.

SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device

Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this

state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the

module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

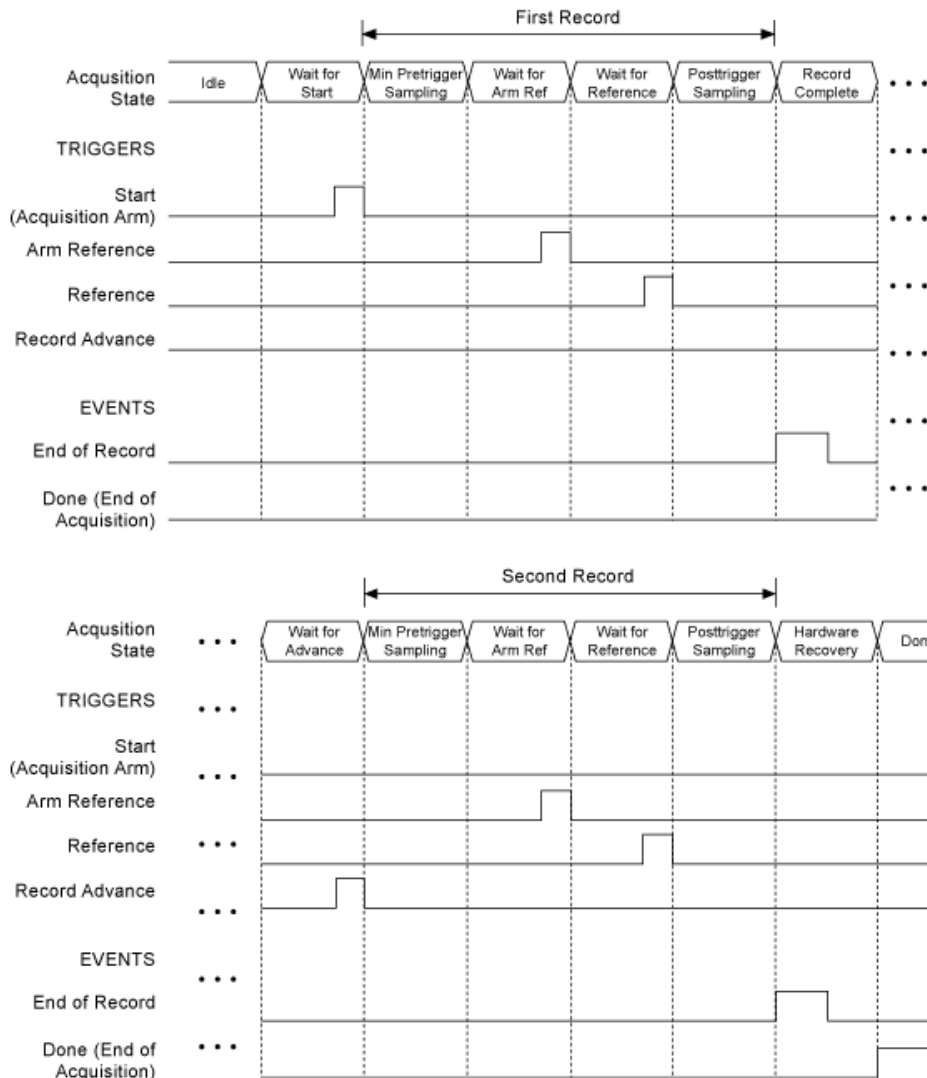
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXIe-5622 Routing Matrix

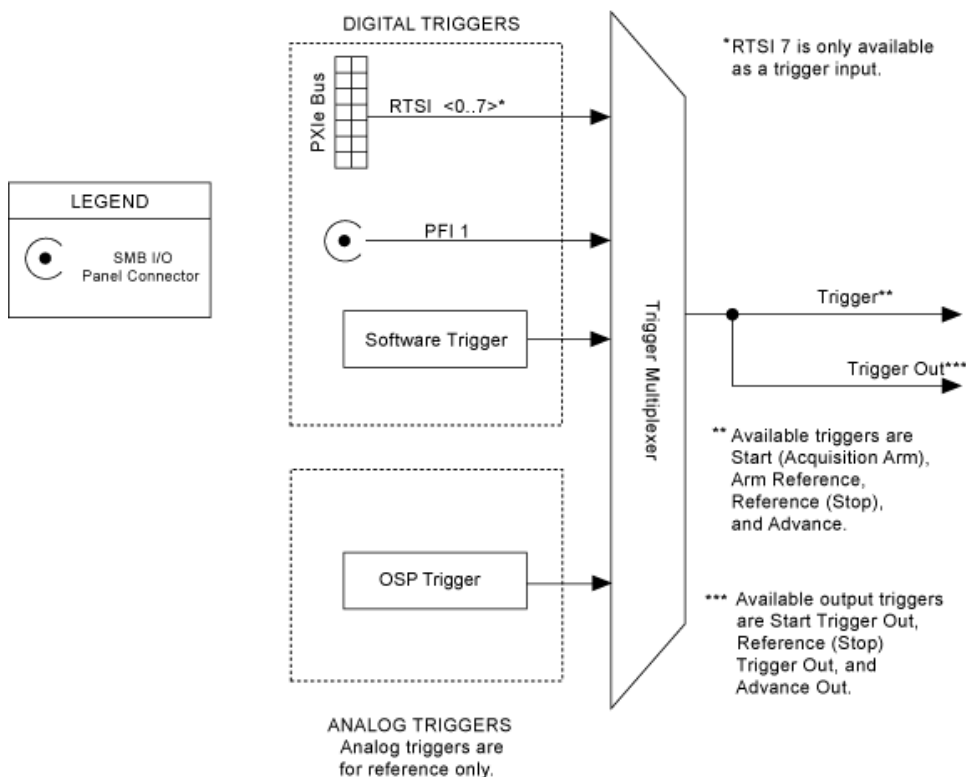
The following table shows the signals available for export from the NI PXIe-5622 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6> (PXI Bus)	CLK OUT (SMA)	PFI 1 (SMB)
Exported Clocks			
Reference Clock (External)	—	√	—
Sample Clock (Internal)	—	√	—
Sample Clock (External)	—	√	—
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

NI 5622 Trigger Types

The NI 5622 supports only [digital triggers](#).

NI PXIe-5622 Trigger Sources



NI 5622 Trigger Holdoff

For the NI 5622, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

NI 5122/5124/5142/5622/5922 Trigger Delay

[Trigger delay](#), which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger

time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

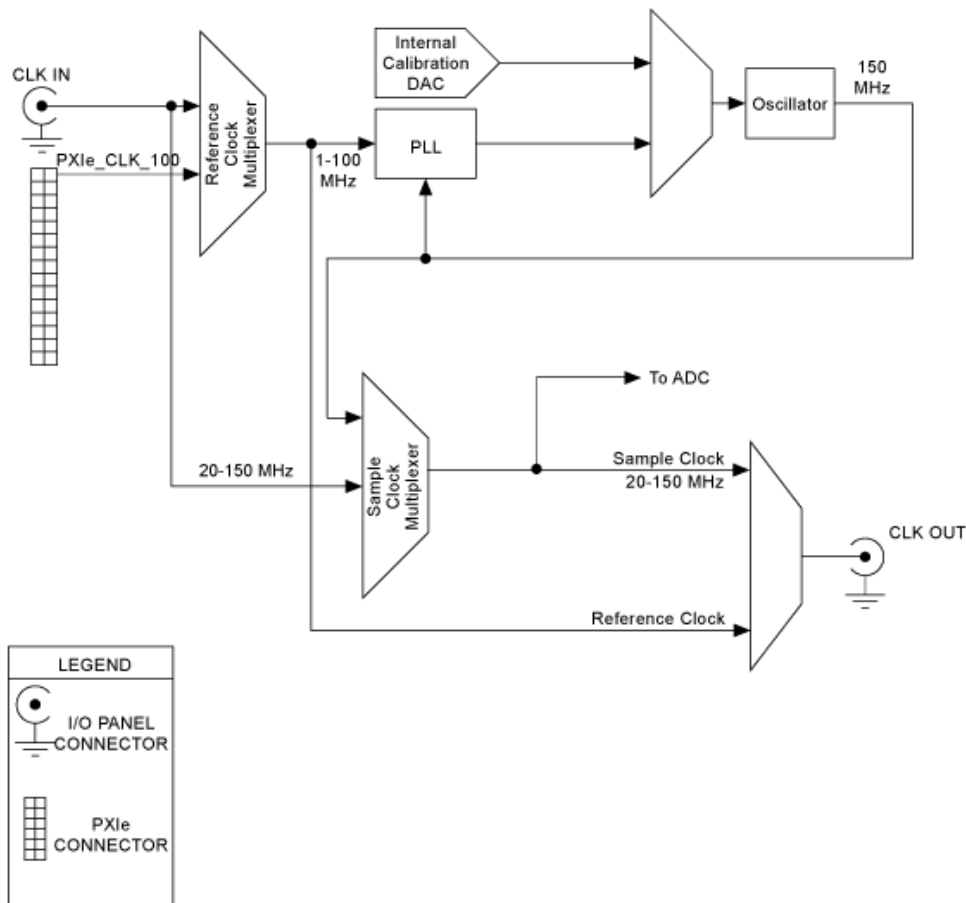
$$\text{Max trigger delay in seconds} = [(2^{34} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock. The maximum trigger delay also changes based on the binary sample width. When binary sample width is set to 8, the maximum trigger delay is doubled. When binary sample width is set to 32, the maximum trigger delay is halved.

NI 5622 Clocking

The clock circuitry on the NI 5622 offers versatile clocking options with the ability to use either the internal 150 MHz sample clock or to accept an external sample clock that you provide. You can also use the phase-locked loop (PLL) circuit on the digitizer to phase lock the internal 150 MHz sample clock with the PXIe 100 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI 5622.



Sample Clock

The sample clock is sent to the ADC and to the input timing engine. The NI 5622 can decimate its sample clocks (internal or external) by an integer divisor. When using an external clock, you can use decimation to achieve rates below the external clock frequency.

Internal Sample Clock

The NI 5622 has an onboard voltage controlled crystal oscillator (VCXO) running at 150 MHz. When using the onboard 150 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is the calibrated 150 MHz frequency of the VCXO. In PLL mode, the device phase locks its 150 MHz sample clock to the supplied reference clock. The PLL mode is useful when synchronizing the digitizer with other devices in a measurement system.

External Sample Clock

Some applications may require sampling at specific intervals that cannot be achieved by using the internal 150 MHz clock. In these cases these digitizers can accept an external sample clock. External clocking also provides another method to synchronize the digitizer to other devices in a measurement system by distributing a common clock to multiple devices. An external sample clock can be supplied to the digitizer from the front panel connector. Refer to the hardware specifications document for external sample clock requirements.

Reference Clock

The reference clock is used in the digitizer's phase-locked loop (PLL) circuit to synchronize the sample clock to the reference clock. The digitizers can accept a reference clock from the front panel (CLK IN) as well as from PXIe_CLK100. This reference clock can be any frequency from 1 MHz to 100 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXIe_CLK100 is always a 100 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the reference clock is `None`, or not to use a reference clock.



Note Locking to a reference clock is not valid when using external sample clocking.

Exporting Sample Clock

To achieve sampling rates other than 150 MS/s when using the internal sample clock, the digitizer decimates the sampled data. When you set a sampling rate of 75 MS/s, the NI 5622 can only store one out of every two samples received from the ADCs to onboard memory. It can sample at rates of $150/n$ MS/s, where **n** is one of the following integer values:


1, 2, 4, 6, 8, 10

12 to 4,096 (Multiples of 4)

4,096 to 8,192 (Multiples of 8)

8,192 to 16,384 (Multiples of 16)

Regardless of decimation, the rate of sampling as seen by the ADC is the only rate that can be exported as sample clock on the CLK OUT connector.

 **Note** The NI 5622 cannot export its sample clock while operating in [DDC](#) mode. If you want to export the sample clock, set the [DDC Enabled](#) property to False (default).

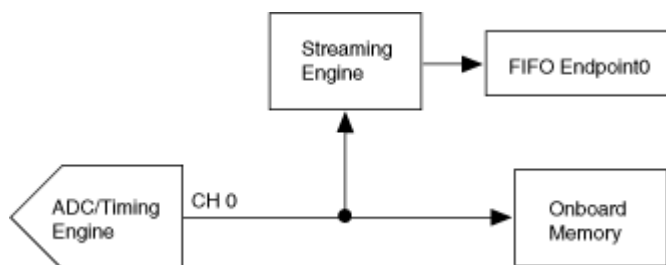
Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI 5622 Routing Matrix](#).

NI PXIe-5622 Peer-to-Peer Streaming

The NI PXIe-5622 supports peer-to-peer streaming using the NI-P2P API. For more information about using NI-P2P to stream data between devices, refer to the [Peer-to-Peer Streaming Help](#). The [NI-SCOPE Function Reference Help](#) and the [NI-SCOPE LabVIEW Reference Help](#) provide more information on configuring the digitizer's endpoint resources to send data to another peer.

The digitizer's single writer endpoint is located on a parallel data path to the onboard memory. Acquired data can be written to this endpoint as well as fetched to the host through onboard memory. By default, data is not written to onboard memory while any peer-to-peer endpoint is enabled. Set the [Onboard Memory Enabled](#) property or the [NISCOPE_ATTR_P2P_ONBOARD_MEMORY_ENABLED](#) attribute to store data to onboard memory while a peer-to-peer endpoint is enabled.



NI 5622 Configuring an Endpoint

To set properties on an endpoint, you must select that endpoint using the [Active Channel](#) property and the endpoint name `FIFOEndpoint0`.

Considerations When Configuring an Endpoint

Keep the following points in mind when configuring an endpoint:

- Set the [P2P Enabled](#) property or the [NISCOPE_ATTR_P2P_ENABLED](#) attribute to TRUE to activate the P2P endpoint and to use any of the P2P features.
- Use the [Channels to Stream](#) property or the [NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM](#) to determine the specific data that is sent to an endpoint.
- Use the [Binary Sample Width](#) property or the [NISCOPE_ATTR_BINARY_SAMPLE_WIDTH](#) attribute to set the sample width.
- Call `niScope Abort`, `niScope Close`, or `niScope Reset` on the digitizer session to reset an endpoint.

On the NI 5622, complex data can be acquired when using the [Onboard Signal Processing](#) block. If the data sent to the endpoint is complex, it will be interleaved by sample, with the I sample followed by the Q sample. If the complex sample is de-interleaved from a U32 element on the FPGA diagram, the I sample will be the lower half and the Q sample will be the upper half.

Configuring a Peer-to-Peer Stream

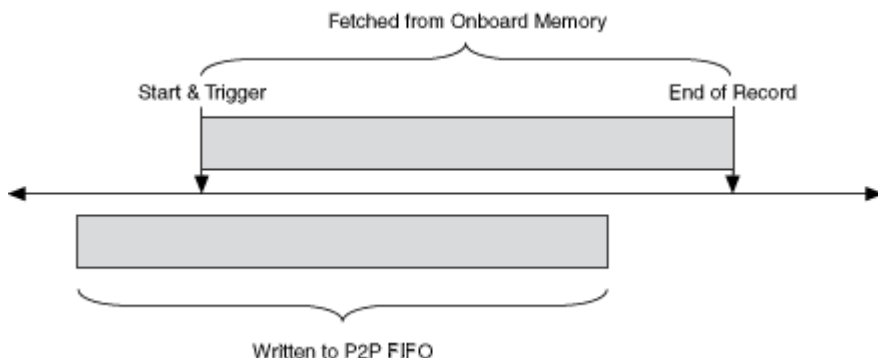
To configure a peer-to-peer stream using the NI-P2P API, a writer and reader handle for each endpoint are required. Use the [niScope Get Stream Endpoint Handle VI](#) to get a writer endpoint handle. Refer to the **Peer-to-Peer Streaming Help** for more information on creating and monitoring peer-to-peer streams.

Configuring a Trigger with a P2P Stream

Depending on the type of trigger configured for the acquisition, data is not written to the P2P endpoints in the same way as it is written to onboard memory.

Immediate Trigger

Because of differences in the data path leading to the P2P endpoints and the onboard memory, the first sample written to the P2P endpoints is from an earlier time than the first sample written to onboard memory.

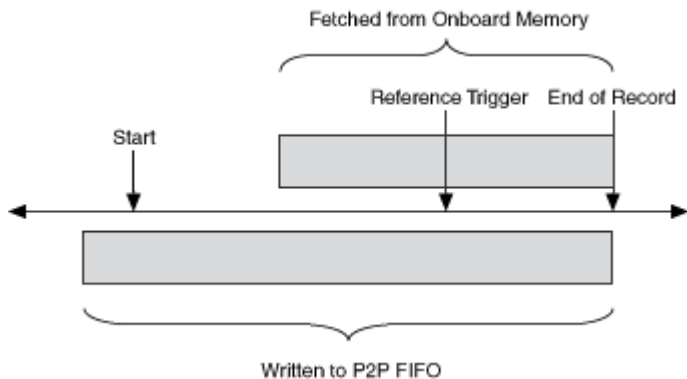


Note The record sent to the onboard memory is the same length as the record sent to the P2P FIFO.

All data written to the P2P endpoints is valid, but the samples are not aligned with the same record stored in onboard memory.

Configured Reference Trigger

When the trigger type is anything other than immediate, the digitizer sends all data between the Start trigger and the End of Record event to the endpoint.



Scaling and Normalization of Binary Data

In a normal acquisition, you cannot assume that the binary data maps perfectly to the vertical range. For example, if you fetch 16-bit binary data, you might expect

that the positive full scale binary value corresponds to the maximum positive voltage of the vertical range, and that the negative full scale binary value corresponds to the maximum negative voltage. This is not the case, however, for a number of reasons.

First, the full scale of the ADC is not mapped to the vertical range. A few codes on the positive and negative ends are left as headroom, in case the input signal slightly exceeds the specified vertical range. Second, the binary values do not account for the required corrections to gain and offset based on calibration data. If you fetch binary data, you can convert to voltage values using the gain and offset values from the `wfmInfo` struct or use the [niScope Get Scaling Coefficients VI](#) or the [niScope_GetScalingCoefficients](#) function to get these coefficients. These values account for the ADC normalization and calibration data.

$$\text{Voltage} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

To get binary data that accounts for both ADC normalization and calibration, use the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function. Applying gain and offset from the normalization coefficients results in a range of binary values that maps exactly to the vertical range you configure.

$$\text{Normalized Binary Data} = (\text{Binary Value} \times \text{Gain}) + \text{Offset}$$

$$\text{Voltage} = \text{Normalized Binary Data} \times (\text{Range} / 2^{\text{resolution in bits}})$$

In a normal acquisition, you must know the gain and offset values to correctly convert the binary data to voltage values. After applying gain and offset values from the [niScope Get Normalization Coefficients VI](#) or the [niScope_GetNormalizationCoefficients](#) function, you can assume the following:

- The maximum positive binary value maps to the maximum positive voltage of the vertical range.
- The maximum negative binary value maps to the maximum negative voltage of the vertical range.
- The vertical range is divided evenly across the possible binary values.

Peer-to-Peer Streaming VIs/Functions

LabVIEW	C/C++
Streaming Session—Use the following NI-P2P VI to configure a peer-to-peer streaming session.	
niScope Get Stream Endpoint Handle VI	niScope_GetStreamEndpointHandle
Scaling and Normalization—Data written to a peer-to-peer endpoint is not scaled or calibrated. Use the following VIs/functions to scale or normalize the binary data.	
niScope Get Wfm Coefficients (poly) VI	niScope_GetNormalizationCoefficients
	niScope_GetScalingCoefficients

Peer-to-Peer Streaming Properties/Attributes

Use the following properties and attributes to configure a peer-to-peer streaming session.

LabVIEW	C/C++
P2P Enabled	NISCOPE_ATTR_P2P_ENABLED
Channels to Stream	NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM
Endpoint Size	NISCOPE_ATTR_P2P_ENDPOINT_SIZE
Samples Available In Endpoint	NISCOPE_ATTR_P2P_SAMPLES_AVAIL_IN_ENDPOINT
Most Samples Available In Endpoint	NISCOPE_ATTR_P2P_MOST_SAMPLES_AVAILABLE_IN_ENDPOINT
Samples Transferred	NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED
Samples Transferred Per Record	NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED_PER_RECORD
Endpoint Overflow	NISCOPE_ATTR_P2P_ENDPOINT_OVERFLOW
FIFO Endpoint Count	NISCOPE_ATTR_P2P_FIFO_ENDPOINT_COUNT
Stream Relative To	NISCOPE_ATTR_P2P_STREAM_RELATIVE_TO

Manual—The following property/attribute activates advanced functionality for a P2P endpoint. To use any of the P2P advanced features, this property/attribute must be set to TRUE. This attribute cannot be used for an endpoint if that endpoint is configured using NI-P2P.

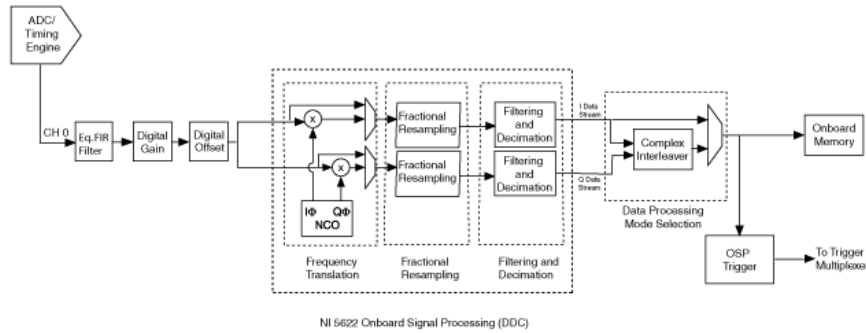
<u>Manual Configuration Enabled</u>	<u>NISCOPE_ATTR_P2P_MANUAL_CONFIGURATION_ENABLED</u>
Configuration—The following properties/attributes provide a way to configure a destination for endpoint data without using NI-P2P. These attributes can be used to send data to a DMA window on a third-party device, or to a memory window on the host machine.	
<u>Data Transfer Permission Address</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR</u>
<u>Data Transfer Permission Address Type</u>	<u>NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR_TYPE</u>
<u>Destination Window Address</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR</u>
<u>Destination Window Address Type</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_TYPE</u>
<u>Destination Window Size</u>	<u>NISCOPE_ATTR_P2P_DESTINATION_WINDOW_SIZE</u>
Notification—The following properties/attributes can be used to send a message corresponding to an event directly to a peer or to a location in host memory.	
<u>Push Message On</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON</u>
<u>Message Push Address</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR</u>
<u>Message Push Address Type</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDRESS_TYPE</u>
<u>Message Push Value</u>	<u>NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VALUE</u>

NI 5622 Onboard Signal Processing (OSP)

The NI 5622 onboard signal processing (OSP) block is a general-purpose block of digital signal processing components.

The following figure shows the main components found in the OSP block of the NI 5622.

Click each area for more information about the main components of the OSP block.



NI 5622 Onboard Signal Processing Components

- [Equalization FIR Filter](#)
- [Digital Gain](#)
- [Digital Offset](#)
- [Fractional Resampling](#)
- [Digital Downconverter \(DDC\)](#)
- [Frequency Translation](#)
- [Filtering and Decimation](#)
- [Data Processing Mode Selection](#)
- [Onboard Signal Processing Triggering](#)

Digital Gain

Digital gain can be used to change the gain of each channel. You can program the digital gain of each channel independently using the [Digital Gain](#) attribute. The gain can range from -1.5 to $+1.5$ (unitless). You can use the digital gain to attenuate the channel data to eliminate overflows in the OSP block. To eliminate overflows, the digital gain should attenuate the ADC data (it should be set to a value less than 1). The digital gain circuit itself can overflow if the following condition is not met:

$$-1 \leq \text{Unity Scaled User Data} \times \text{Pre-Filter Gain} \leq 1$$

If an overflow occurs, the data is clipped and NI-SCOPE returns an error. To prevent data clipping, attenuate the waveform data or reduce the digital gain.



Tip To change the default error reporting caused by OSP overflows, change the [Overflow Error Reporting](#) attribute.

You can also use Digital Gain to do fine-gain adjustments on the ADC data to compensate for channel or system gain imperfections. These adjustments occur in hardware before the data is stored to memory, which saves an additional processing step that would normally occur in software.

Digital Offset

The digital offset can be used to change the offset of each channel. You can program the digital offset of each channel independently using the [Digital Offset](#) attribute. The offset can range from $-(\mathbf{Vertical\ Range} \times 0.4)$ to $+(\mathbf{Vertical\ Range} \times 0.4)$. The digital offset circuit can overflow if the following condition is not met:

$$-(\mathbf{Vertical\ Range}/2) \leq \mathbf{Channel\ Data} + \mathbf{Digital\ Offset} \leq +\mathbf{Vertical\ Range}/2.$$

If an overflow occurs, the data is clipped and NI-SCOPE returns an error. To prevent data clipping, attenuate the waveform data or reduce the digital offset.



Tip To change the default error reporting caused by OSP overflows, change the [Overflow Error Reporting](#) attribute.

Fractional Resampling

Fractional resampling provides higher OSP sample rate resolution. Typical demodulation schemes require data to be sampled at an exact sample rate. This was traditionally accomplished using one of the following methods:

- Using an external sample clock.
- Fractional resampling of the data in software.

The Fractional Resampling stage of the OSP block allows you to move the fractional resampling processing to hardware without using an external clock. Use the [Fractional Resample Enabled](#) property or the [NISCOPE_ATTR_FRACTIONAL_RESAMPLE_ENABLED](#) attribute to enable fractional resampling.

Equalization FIR Filter

Equalization FIR Filter is a generic FIR filter. You can download coefficients to the filter using the [niscope Configure Equalization Filter Coefficients VI](#) or the [niScope_ConfigureEqualizationFilterCoefficients](#) function.

The number of coefficients you can download to the filter is determined by the [Equalization Num Coefficients](#) property or the [NISCOPE_ATTR_EQUALIZATION_NUM_COEFFICIENTS](#) attribute. After the coefficients are downloaded, you can use the [Equalization Filter Enabled](#) property or the [NISCOPE_ATTR_EQUALIZATION_FILTER_ENABLED](#) attribute to enable the filter in the hardware. The Equalization FIR Filter can overflow if the data values exceed the vertical range.

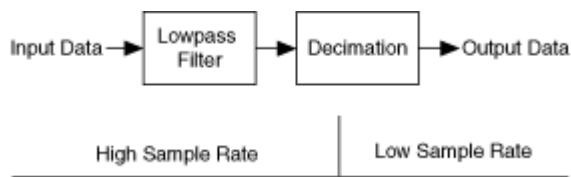
Frequency Translation

Frequency translation moves the [Nyquist](#) frequency spectrum of the acquired signal by the amount specified by the [Center Frequency](#) attribute. This is accomplished by multiplying the acquired data stream with the output of the numerically controlled oscillator (NCO). The NCO is a digital circuit that creates two sine waves of the same frequency (the Center Frequency) with two independent phases. You can use the Center Frequency attribute to set the carrier frequency. The I phase cosine waveform is multiplied by the I signal path, and the Q phase cosine waveform is multiplied by the Q data path. The I and Q phases are programmable from 0° to 360° by setting the [Frequency Translation Phase I](#) and [Frequency Translation Phase Q](#) attributes/properties. Both the frequency and the phases can be updated during acquisition. You can consider the values of the Frequency Translation Phase I/Q attributes as the start phase of the NCO waveforms. The NCO outputs start at the specified phase values when the start trigger is received. Normally, the I and Q phases would be set to 0° and 90°, respectively, but the flexibility allows you to correct any externally-induced impairments in the IF signal.

Filtering and Decimation

The filtering and decimation stage of the OSP block reduces the effective sample rate of the digitizer while protecting the frequency spectrum of the decimated data from aliases. This protection occurs when the data passes through a lowpass filter

before decimation. The normal decimation in digitizers (when not using OSP) does not protect the frequency spectrum of the decimated data from aliases.



Data Processing Mode Selection

The Data Processing Mode Selection stage selects between Real and Complex processing. In Real mode, only the Real I data is returned when a waveform fetch is executed. In Complex mode, the I and Q data streams are interleaved before they are stored to memory. When the complex data is fetched, the data is returned as interleaved I and Q data. The data can be fetched as an array of complex IQ samples, as an array of interleaved real I and Q samples, or as separate arrays of real I and Q samples.

Onboard Signal Processing Triggering

In [DDC](#) mode, the NI 5142 has the ability to analog trigger from circuitry located on the daughtercard, or from the OSP circuit located at the output of the DDC. The analog circuit on the daughtercard always triggers on data before any onboard signal processing occurs. The OSP circuit always triggers on data after onboard signal processing occurs.

In [DDC](#) mode, the NI 5622 has the ability to analog trigger only from the OSP circuit located at the output of the DDC. The OSP circuit always triggers on data after onboard signal processing occurs.

When the DDC is in [complex mode](#), the OSP circuit triggers on the magnitude of both the I stream and the Q stream. The circuit uses the following formula:

$$\text{Sqrt}(I^2 + Q^2)$$

This magnitude is proportional to the power of the signal. By default, the device uses the analog circuit on the daughtercard. You can change the trigger used by the

device by using the [Ref Trigger Detector Location](#) property or the `NISCOPE_ATTR_REF_TRIGGER_DETECTOR_LOCATION` attribute.

The OSP trigger circuit can also trigger on the beginning of [bursts of data](#). To enable this functionality, configure the [Ref Trigger Minimum Quiet Time](#) property or the `NISCOPE_ATTR_REF_TRIGGER_MINIMUM_QUIET_TIME` attribute to the value of time that corresponds to the amount of time that the signal must be below the trigger level to constitute a quiet time. For more information, refer to [Burst Triggers](#).

OSP Basic Properties/Attributes

The following topics contain overviews of the basic OSP properties/attributes.

- [Center Frequency](#)
- [Data Processing Mode](#)
- [DDC Enabled](#)
- [Frequency Translation Enabled](#)
- [Q Source](#)
- [Frequency Translation Phase I](#)
- [Frequency Translation Phase Q](#)
- [Overflow Error Reporting](#)
- [Digital Gain](#)
- [Digital Offset](#)

DDC Enabled

The DDC Enabled property/attribute activates the functionality of the DDC block. To use any of the features in the DDC block, you must set DDC Enabled to TRUE.

LabVIEW	C/C++
DDC Enabled	<code>NISCOPE_ATTR_DDC_ENABLED</code>



Tip To achieve maximum flatness in the filter response of the device in DDC mode, set the maximum input frequency to full bandwidth (-1). However, keep in mind that this action does not protect against aliasing frequency content of the input signal above $(0.5 \times \text{Sample})$

Clock Timebase Rate). When using internal clocking, the Sample Clock Timebase Rate is 100 MS/s (NI 5142) or 150 MS/s (NI 5622).

Hardware Calibration of Binary Data

In a normal acquisition, you cannot assume that the binary data maps perfectly to the vertical range. For example, if you fetch 16-bit binary data, you might expect that the positive full scale binary value corresponds to the maximum positive voltage of the vertical range, and that the negative full scale binary value corresponds to the maximum negative voltage. This is not the case, however, for a number of reasons. First, the full scale of the ADC is not mapped to the vertical range. A few codes on the positive and negative ends are left as headroom, in case the input signal slightly exceeds the specified vertical range. Second, the binary values do not account for the required corrections to gain and offset based on calibration data. If you fetch binary data, you can convert to voltage values using the gain and offset values from the `wfmInfo` struct. These values account for the ADC normalization and calibration data.

$$\text{Voltage} = (\text{Binary Value} \times \text{wfmInfo.gain}) + \text{wfmInfo.offset}$$

When the DDC is enabled, the onboard signal processing block accounts for both ADC normalization and calibration data. So the waveform samples are stored and can be fetched as calibrated, normalized binary data. This means that the range of binary values maps exactly to the vertical range you configure.

$$\text{Voltage} = \text{Binary Value} \times \text{Range} / 2^{(\text{ADC resolution in bits})}$$

The gain and offset values in the `wfmInfo` struct work just as well in this case (but they do not hold normalization and calibration information), so the following conversion still works:

$$\text{Voltage} = (\text{Binary Value} \times \text{wfmInfo.gain}) + \text{wfmInfo.offset}$$

In a normal acquisition, you must know the gain and offset values to correctly convert the binary data to voltage values. In DDC mode, you can assume the following:

- The maximum positive binary value maps to the maximum positive voltage of the vertical range.

- The maximum negative binary value maps to the maximum negative voltage of the vertical range.
- The vertical range is divided evenly across the possible binary values.

These assumptions may be useful for streaming or logging binary data, because you do not need to know the gain and offset value to recover the measured voltages.

Center Frequency

The Center Frequency property/attribute indicates the frequency by which the frequency translation stage of the DDC block frequency translates the input data; that is, the center frequency of the region of the spectrum that you want to downconvert to baseband.

LabVIEW Property	C/C++ Attribute
Center Frequency	NISCOPE_ATTR_CENTER_FREQUENCY

Frequency Translation Enabled

The Frequency Translation Enabled property enables frequency translating the data around the user-selected center frequency down to baseband.

LabVIEW Property	C/C++ Attribute
Frequency Translation Enabled	NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_ENABLED

Data Processing Mode

The Data Processing Mode property/attribute determines how the DDC block processes data. If Data Processing Mode is set to Real, only the I data stream is enabled and the waveform data points are real numbers (I data). If Data Processing Mode is set to Complex, both the I and Q data streams are enabled and interleaved, and waveform data points are complex numbers (IQ data).

LabVIEW Property	C/C++ Attribute
Data Processing Mode	NISCOPE_ATTR_DATA_PROCESSING_MODE

Frequency Translation Phase I

The Frequency Translation Phase I property/attribute specifies the I oscillator phase of the sine wave in degrees at the first point acquired.

LabVIEW Property	C/C++ Attribute
<u>Frequency Translation Phase I</u>	<u>NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHASE_I</u>

Frequency Translation Phase Q

The Frequency Translation Phase Q property/attribute specifies the Q oscillator phase of the sine wave in degrees at the first point acquired. Use this property only when the Data Processing Mode is set to Complex.

LabVIEW Property	C/C++ Attribute
<u>Frequency Translation Phase Q</u>	<u>NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHASE_Q</u>

Overflow Error Reporting

The Overflow Error Reporting property/attribute configures error reporting when the DDC block detects an overflow in any of its stages. All overflows lead to clipping of the waveform.

LabVIEW Property	C/C++ Attribute
<u>Overflow Error Reporting</u>	<u>NISCOPE_ATTR_OVERFLOW_ERROR_REPORTING</u>

Q Source

The Q Source property/attribute indicates the channel that is the input of the Q data stream of the DDC.

LabVIEW Property	C/C++ Attribute
<u>Q Source</u>	<u>NISCOPE_ATTR_DDC_Q_SOURCE</u>

Digital Gain

The Digital Gain property/attribute applies gain to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

$$(\text{ADC value} \times \text{digital gain}) + \text{digital offset}$$

LabVIEW Property	C/C++ Attribute
<u>Digital Gain</u>	<u>NISCOPE_ATTR_DIGITAL_GAIN</u>

Digital Offset

The Digital Offset property/attribute applies offset to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

$$(\text{ADC value} \times \text{digital gain}) + \text{digital offset}$$

LabVIEW Property	C/C++ Attribute
<u>Digital Offset</u>	<u>NISCOPE_ATTR_DIGITAL_OFFSET</u>

Common DDC Applications (NI 5622)

The DDC block of the NI 5622 is useful for the following applications:

- [Alias-Protected Decimation](#)
- [AM Demodulation](#)
- [Quadrature Downconversion](#)

To use the NI 5622 in normal digitizer mode, set the [DDC Enabled](#) property to False (default).

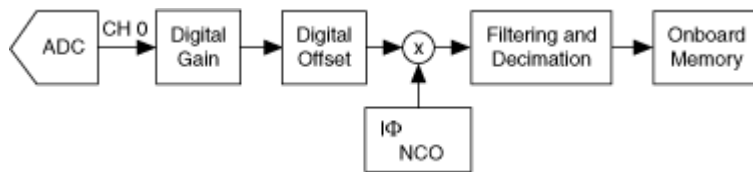
Alias-Protected Decimation



For alias-protected decimation applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Real.
3. Set the [Min Sample Rate](#) property to the desired decimated sample rate.
4. Set the [Frequency Translation Enabled](#) property to False.

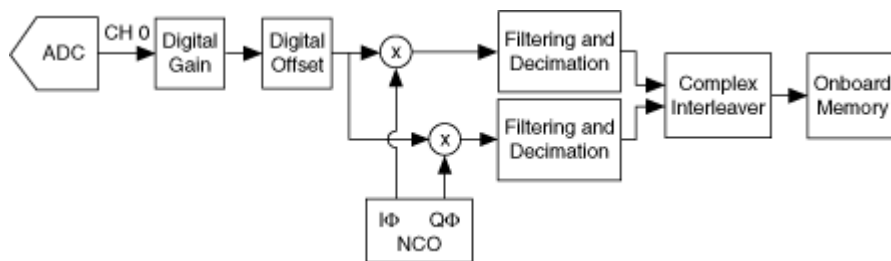
AM Demodulation



For AM demodulation applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Real.
3. Set [Min Sample Rate](#) property to the desired baseband sample rate.
4. Set [Frequency Translation Enabled](#) property to True.
5. Set [Center Frequency](#) property to the carrier frequency of the AM signal you want to downconvert.

Quadrature Downconversion



For quadrature downconversion applications, complete the following steps:

1. Set the [DDC Enabled](#) property to True.
2. Set the [Data Processing Mode](#) property to Complex.

3. Set the [Min Sample Rate](#) property to the desired IQ data rate.
4. Set the [Frequency Translation Enabled](#) property to True.
5. Set the [Center Frequency](#) property to the desired center frequency of the IF band you want to downconvert.

NI 5622 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time exceed those specified, and your application requires tight specifications, calibration is required.

For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of input} + 10 \text{ mV})$, and you apply 5 V to the input, the error is:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{-}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple integrated instruments, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors.

If the ambient temperature is outside of the 18-28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18-28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming and expensive and is infrequently done, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$\text{TC} = (0.1\% \text{ of input} + 1 \text{ mV}) / ^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / ^\circ\text{C)}$$

The additional error is

$$20 \text{ }^\circ\text{C} \times \text{TC} = \pm(2\% \text{ of input} + 20 \text{ mV}) \text{ or } 120 \text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function.

For the NI 5622 this self-calibration capability yields the following benefits:

- Corrects for gain errors within the digitizer by comparison to a precision, high-stability internal sine wave reference. This is done for all ranges, and both filter paths (enabled/disabled).
- Calibrates trigger timing, as well as the time-to-digital conversion ([TDC](#)) circuitry to ensure accurate trigger timing and time-stamping.
- Takes approximately 1 minute to complete.



Note The analog input impedance changes from 50 Ω to high impedance during self-calibration as input is switched from the IF IN connector to an internal calibration signal.


When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or 90 days after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and one-year calibration cycle for AC gain and frequency response. When the calibration interval expires, an external calibration is required.

The NI 5622 has a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

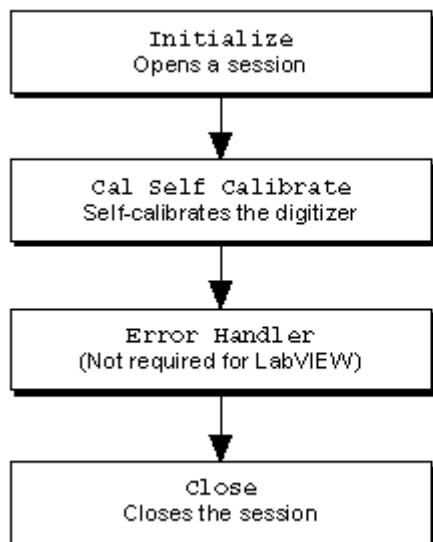
Input Connections During Self-Calibration

The NI 5622 internal circuitry is automatically isolated from the input during self-calibration. However, as a precautionary measure, always disconnect the inputs as directed.

 **Note** The input impedance changes at the IF IN connector from 50 Ω to high impedance during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



NI-SCOPE provides the `Calibrate` example, which you can find by using the shortcut at `Start»All Programs»National Instruments»NI-SCOPE»Examples`.

Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
-------------	--------	------	-------

External calibration	Calibrate time drift of onboard reference Gain	Every year	Calibrates and verifies to full specifications
Self-calibration	Gain Trigger timing	90 days, or when temperature changes >5 °C	Ensures amplitude accuracy Ensures trigger accuracy
No calibration	None, within 1 year calibration cycle or if temperature stays within ±5 °C	High absolute accuracy not required outside of 5 °C	If self-calibration is not used, derate the absolute accuracy using the specified Tempco

SMC-Based Digitizers Multiple-Record Acquisition

SMC-based digitizers support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Digitizers Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Digitizers Acquisition Engine State Diagram](#).

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the digitizer, NI-SCOPE limits to approximately

100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5622 Onboard Memory

The NI 5622 allocates at least 512 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. The minimum size for a buffer in the onboard memory can be determined by the following formula.

Minimum Space Allocated for Record = (**Record Length** × **N** bytes/sample) + **M** bytes

where

N = 2 for Real mode

N = 4 for Complex mode

M = 780 for Real mode

M = 960 for Complex mode

Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by the minimum space allocated per record. Refer to the hardware specifications document for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5622 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

NI 5900

The NI PXI-5900 is a differential amplifier designed for use with NI high-speed digitizers, particularly the [NI 5922](#) high-resolution digitizer.

Features

- Two differential input channels
- AC, DC, and GND input coupling
- 1 M Ω or 50 Ω software-selectable input impedance
- PXI form factor


Programming Options

- Use with an NI digitizer as part of an NI-SCOPE session (recommended); refer to the [niScope Initialize with Options VI](#) or the [niscope_InitWithOptions](#) function for programming details.



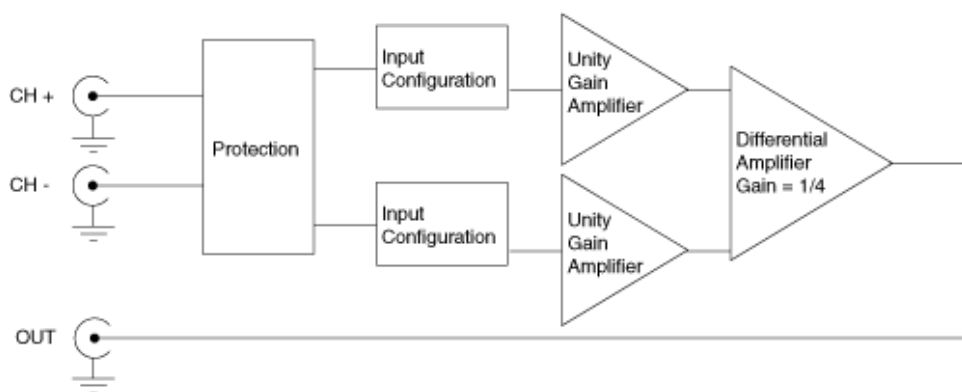
Note In this mode, waveform data returned by the NI-SCOPE session will be automatically scaled to account for the gain and offset of the 5900.

- Use as a standalone session by opening an NI-SCOPE session using the device name of the NI 5900.

 **Note** In this mode, waveform data returned by the NI-SCOPE session will **not** be scaled to account for the gain and offset of the 5900. You must scale your waveform data by the correct gain and offset after you have fetched it from the device. To determine the calibrated gain and offset for the 5900, query the [Accessory Gain](#) and [Accessory Offset](#) properties or the `NISCOPE_ATTR_SIGNAL_COND_GAIN` and `NISCOPE_ATTR_SIGNAL_COND_OFFSET` attributes.

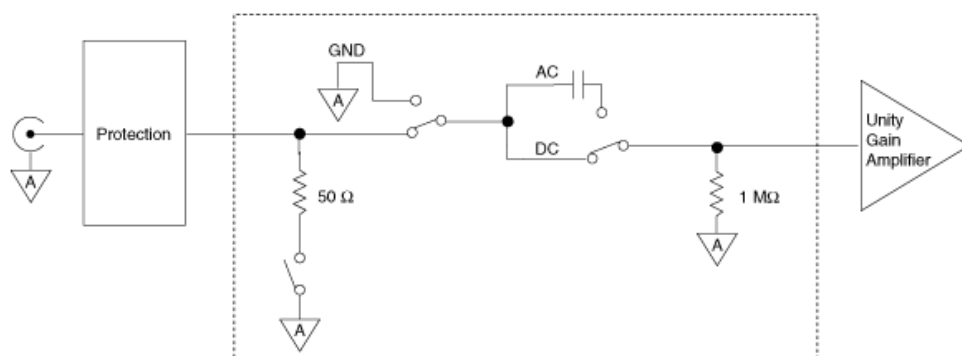
NI 5900 Block Diagram

The following figure shows a detailed block diagram of the NI 5900 differential amplifier.



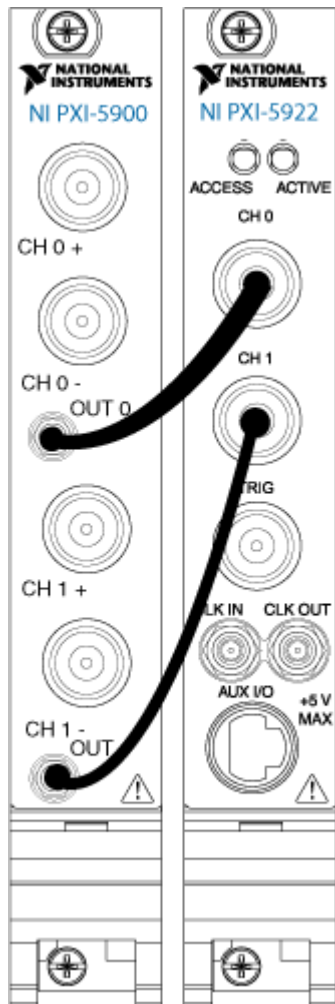
NI 5900 Input Configuration

The following figure shows the input configuration of the NI 5900 differential amplifier.



NI 5900 Front Panel

The following figure shows the front panel of the NI PXI-5900. Descriptions of the connectors are shown below



Connectors

NI PXI-5900 accessories have the following six connectors on the front panel.

Connector	Description	Function
CH 0+	Standard BNC connector	Differential analog input signal for channel 0
CH 0-	Standard BNC connector	Differential analog input signal for channel 0

CH 0 OUT	SMB jack	Single-ended analog output for channel 0; approximately $((CH0+) - (CH0-)) / 4$
CH 1+	Standard BNC connector	Differential analog input signal for channel 1
CH 1-	Standard BNC connector	Differential analog input signal for channel 1
CH 1 OUT	SMB jack	Single-ended analog output for channel 1; approximately $((CH1+) - (CH1-)) / 4$

NI 5922

The NI PXI/PCI-5922 is a 2-channel, high-resolution digitizer offering state-of-the-art resolution for sampling rates up to 15 MS/s.

Features

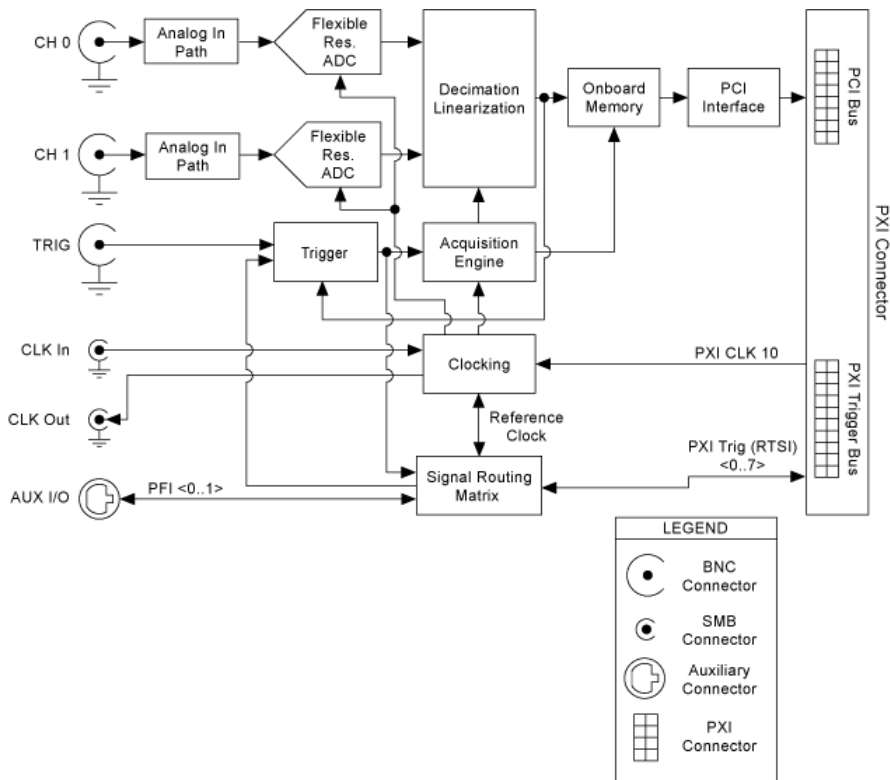
- Two channels
- [Flexible resolution](#) (16 to 24 bits)
- Extremely low noise
- Excellent linearity
- 1 M Ω or 50 Ω selectable input impedance
- Analog and digital triggering
- Three memory options: 8, 32, or 256 MB per channel
- [NI-TClk synchronization](#)

Related Topics

[Features Supported by Device](#)

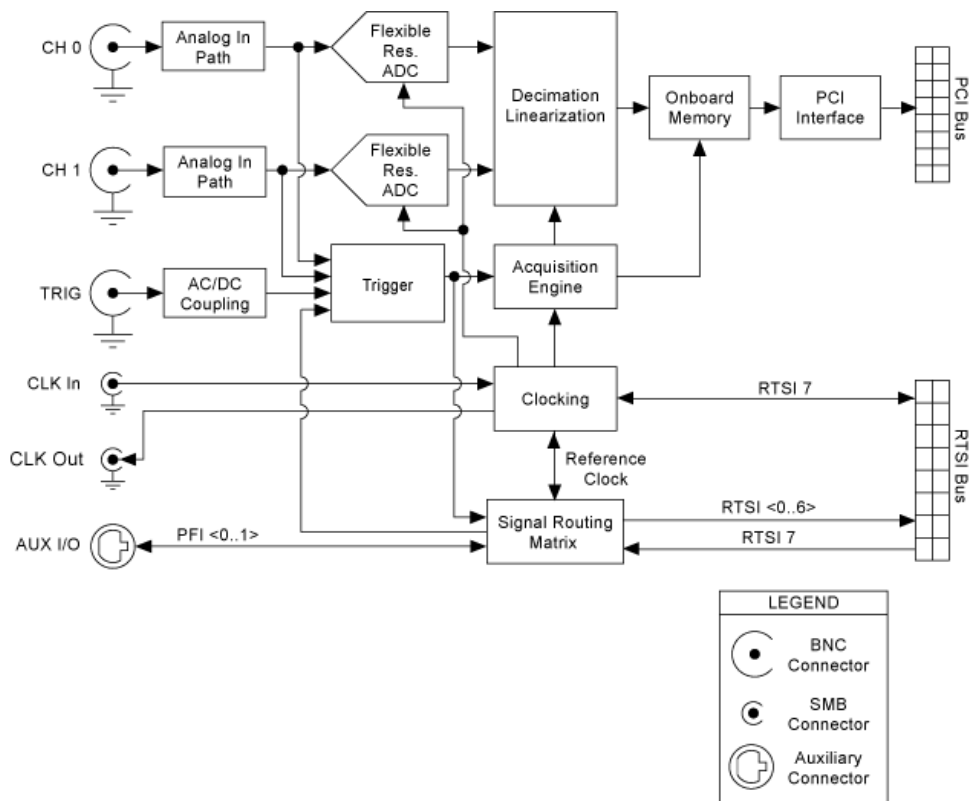
NI PXI-5922 Block Diagram

The following figure shows a detailed block diagram of the NI PXI-5922.



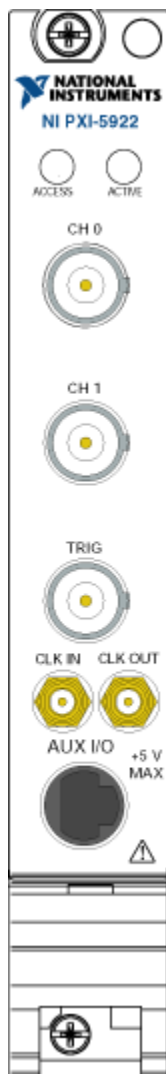
NI PCI-5922 Block Diagram

The following figure shows a detailed block diagram of the NI PCI-5922.



NI PXI-5922 Front Panel

The following figure shows the front panel connectors of the NI PXI-5922.



LEDs

The NI PXI-5922 has two LEDs to indicate status: Access and Active.

Access LED


The Access LED indicates basic hardware status, as listed in the following table.

Color	Indications
Off	Module is not yet functional, or the module has detected a problem with a power rail.
Amber	The module is being accessed.

Green	The module is ready to be programmed by NI-SCOPE.
-------	---

Active LED

The Active LED indicates the module state, as listed in the following table.

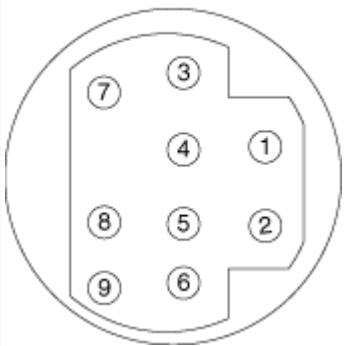
Color	Indications
Off	Module is not armed, triggered, or acquiring a waveform.
Amber	The module is armed and waiting for a Reference (Stop) Trigger.
Green	The module has received a Reference (Stop) Trigger. Also indicates that the module is acquiring a waveform.
Red	<p>The module has detected an error. NI-SCOPE must access the module to determine the cause of the error. The LED remains red until the error condition is removed. Example errors include the following:</p> <ul style="list-style-type: none"> ▪ PLL unlocked—The module has detected an unlocked condition on a previously locked phase-locked loop (PLL). A PLL that is unlocked while in reset does not show an error. ▪ External sample clock error—the module is unable to detect the external sample clock. ▪ The module has detected an overheating error. Refer to Thermal Shutdown for more information. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note Certain driver interactions may cause the Active LED to flash red. An error condition does not exist unless the Active LED remains red.</p> </div>


Connectors

The NI PXI-5922 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions
TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

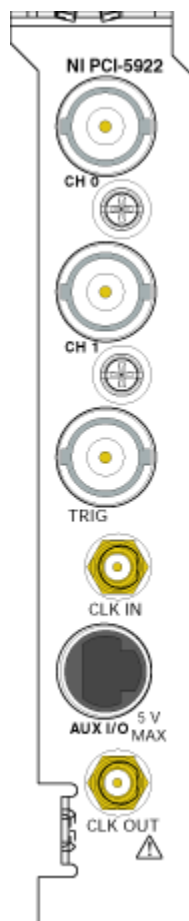
AUX I/O Connector Pin Assignments

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI PCI-5922 Front Panel

The following figure shows the front panel of the NI PCI-5922. Descriptions of the connectors are shown below.



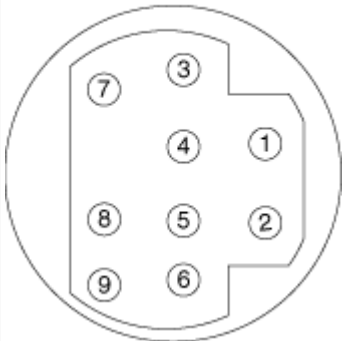
Connectors


The NI PCI-5922 has the following six connectors on the front panel.

Connector	Description	Function
CH 0, CH 1	Standard BNC connector	Analog input connection; digitizes data and triggers acquisitions

TRIG	Standard BNC connector	External analog trigger connection; signals on the TRIG connector cannot be digitized
CLK IN	SMB jack	Imports an external reference clock to the digitizer
CLK OUT	SMB jack	Exports the digitizer reference clock
AUX I/O	9-pin mini-circular DIN connector	Provides access to the external digital trigger lines, PFI 0 and PFI 1 (with optional cable)

AUX I/O Connector Pin Assignments

9-Pin DIN Connector	Pin Assignments and Connector Descriptions	
	Pin Number	Description
	1	5 V (Fused)
	2	GND
	3	Reserved
	4	Reserved
	5	Reserved
	6	PFI 1
	7	Reserved
	8	Reserved
	9	PFI 0

 **Note** Be sure to use an NI adapter cable or a cable that has the same pinout shown in the previous figure.

NI 5922 Flexible Resolution

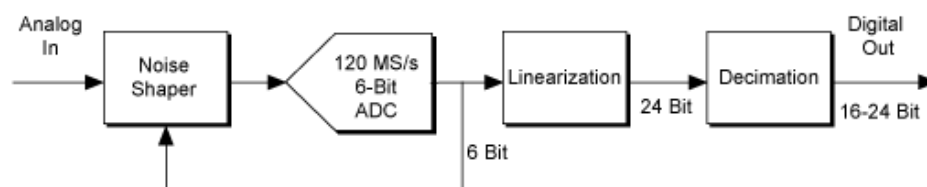
Flexible resolution is a National Instruments proprietary technology for a very high-resolution analog-to-digital converter (ADC). The Flex ADC trades off resolution for speed and provides the best possible resolution over a wide range of sampling

rates. The following table shows the resolution of the NI 5922 for different sampling rates.

Sample Rate	Resolution	Bandwidth ($0.4 \times$ Sample Rate)
100 kS/s	24 bit	40 kHz
1 MS/s	22 bit	400 kHz
5 MS/s	20 bit	2 MHz
10 MS/s	18 bit	4 MHz
15 MS/s	16 bit	6 MHz

How Flexible Resolution Works

The NI 5922 uses an internal 6-bit 120 MS/s ADC sourced through a noise-shaping circuit that moves quantization noise on the output of the ADC from lower frequencies to higher frequencies. Data from the ADC is then corrected in real time in a digital linearization block to an accuracy of 24 bits, which corrects for static as well as dynamic errors in the ADC. The linearization block is configured during self-calibration. A digital lowpass filter applied to the data from the linearization block removes all but a fraction of the original shaped quantization noise. The signal is then resampled to a lower sampling frequency and a higher resolution. Flexible resolution provides antialiasing protection due to the digital lowpass filter.



Why Errors Can Occur During Acquisition

Like any other type of converter that uses noise shaping to enhance resolution, the frequency response of the converter is only flat to its maximum useful bandwidth. The NI 5922 has a maximum bandwidth of 6 MHz. Beyond this frequency, there is a span where the converter acts resonant and where a signal is amplified before being converted. These signals are attenuated in the subsequent digital filter to prevent

aliasing. However, if the applied signal contains major signal components in this frequency range, such as harmonics or noise, the converter may overload and signal data may become invalid. In this case, you receive an overload error message. You must then either select a higher input range or attenuate the signal.



Note If you exceed the device input range, you receive an error message and your data may become invalid.

Thermal Shutdown

NI-SCOPE driver software supports thermal shutdown capability for SMC-based digitizers. This capability allows the digitizer to detect when it has reached an over temperature range condition, and to then power down, preventing damage to the digitizer.

Air circulation paths, fan settings, and space allowances are several factors that can influence device temperature. To prevent thermal shutdown, follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** document that shipped with your device. Refer to the specifications document for your device to find the correct operating temperature range.

In the event that the digitizer powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, for example, the commit function and the self-calibration function.
- Measurement & Automation Explorer (MAX) returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer and the chassis that contains the digitizer (required for the NI PXIe-5185/5186).
- For the NI PXIe-5160/5162, you can disable and enable the device from the Device Manager.

- For PXI and PCI devices, you can call `niScope_ResetDevice` or perform a device reset in MAX. For more information on resetting a device in MAX, select Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx within MAX.

The thermal shutdown error continues to be reported until the device is successfully reset.

Notes

- When overheating occurs on the NI PXI-5122, the Active LED turns red and the Access LED may turn red, depending on the hardware revision you have.
- When overheating occurs on the NI PXI-5124/5142/5152/5153/5154/5922 or the NI PXIe-5122/5185/5186/5622, the Active LED turns red.
- For the NI 5185/5186, you must follow the guidelines described in the **Maintain Forced-Air Cooling Note to Users** to prolong the life of the device. If the NI 5185/5186 goes into thermal shutdown, a power cycle is required to re-enable the device. You can read the current temperature of the device using LabVIEW by selecting the Device Temperature property in the NI-SCOPE property node. It is recommended that this value stay below 100 °C to maximize the life of the NI 5185/5186. The NI 5185/5186 will go into thermal shutdown when its temperature exceeds 107 °C.

Related Topics

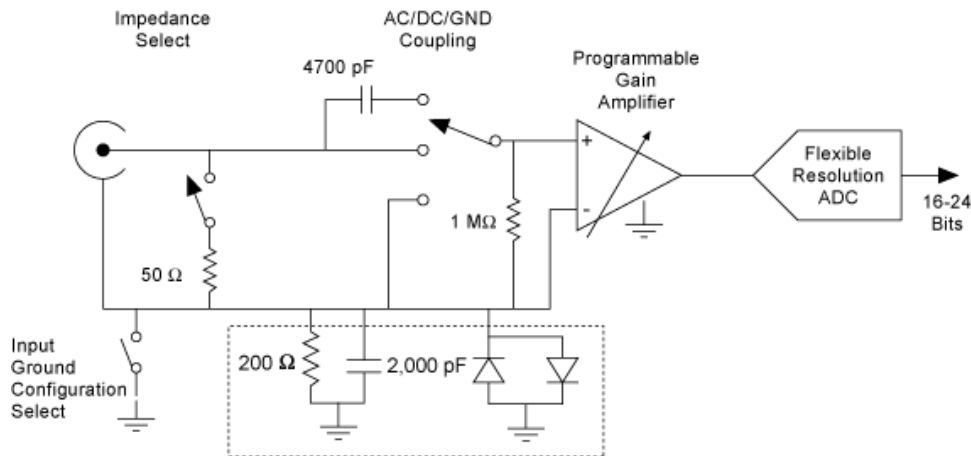
[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5922 Input Signal Conditioning

The NI 5922 provides two independent input channel signal conditioning paths. Each path has the following options:

- 50 Ω input impedance or 1 M Ω input impedance
- AC, DC, or GND coupling
- 2 V_{pk-pk} or 10 V_{pk-pk} input range
- Unbalanced differential or single-ended input ground configuration

The following diagram illustrates the input signal conditioning paths.

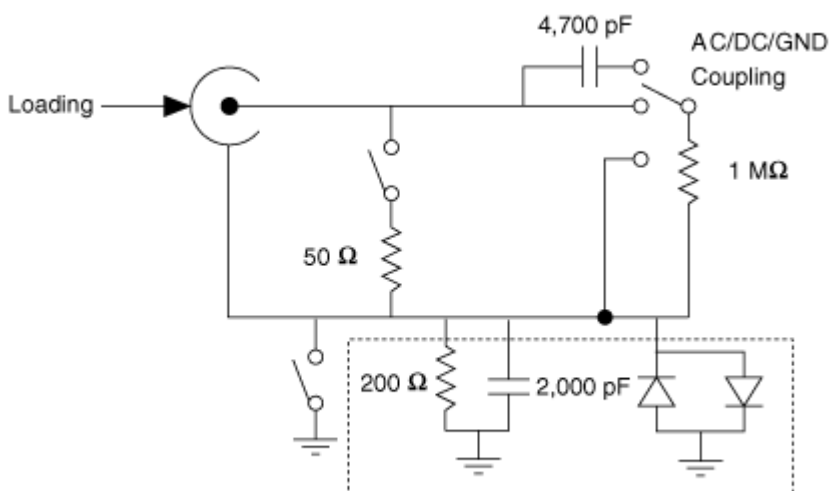


Note The ground on the device inputs is connected to the chassis ground. For information regarding safety considerations when grounding, such as maximum voltage between the device inputs and the device ground, refer to the "Safety" section of the device specifications.

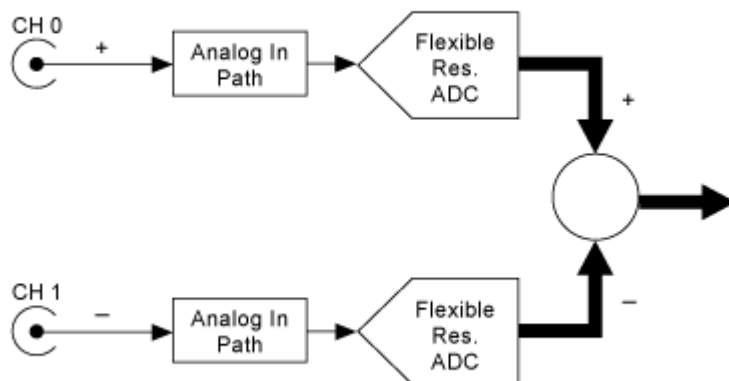
NI 5922 Channel Terminal Configuration

The NI 5922 provides the following three options for channel terminal configuration:

- **Single-ended**—In this mode, the shield of the BNC connector is grounded directly to the digitizer front panel.
- **Unbalanced differential**—This mode is sometimes referred to as pseudodifferential. In this mode, the shield of the input BNC is connected to ground through a resistor. In parallel with the resistor, two antiparallel diodes have been added to protect the programmable gain amplifier, which is limited to ± 5 V signals. This mode allows the differential programmable gain amplifier to reject common-mode noise sources from the input signal.



- **Differential**—The difference between the two channels is derived in hardware by converting both channels to digital and subtracting the outputs digitally as (CH 0 – CH 1). In differential mode, the two channels are configured as unbalanced differential. CH 1 automatically uses the configuration of CH 0. This mode enables one channel of balanced differential measurements.





Note To set the channel terminal configuration, use the [Channel Terminal Configuration](#) property or the `NISCOPE_ATTR_CHANNEL_TERMINAL_CONFIGURATION` attribute.

NI 5922 Input Ranges

The NI 5922 has the following input ranges:

- $\pm 1\text{ V}$ ($2\text{ V}_{\text{pk-pk}}$)
- $\pm 5\text{ V}$ ($10\text{ V}_{\text{pk-pk}}$)

NI 5922 Input Impedance

You can set the NI 5922 analog input impedance to either $50\ \Omega$ or $1\ \text{M}\Omega$.

Input Bias Current

The input buffer on the NI 5922 is implemented with bipolar amplifiers for the lowest possible input noise. Compared to J-FET amplifiers, the bipolar implementation has higher input bias current. The NI 5922 contains a circuit that compensates for bias current. The bias compensation is adjusted during self-calibration. Due to the granularity of the adjustment and temperature drift, the bias current cannot be completely eliminated.

When the NI 5922 is sourced from a very high-impedance source (or left floating) and input impedance is set to $1\ \text{M}\Omega$, the bias current may result in substantial DC offset.

Protection

The input amplifier on the NI 5922 contains a protection circuit that protects the circuitries from damage caused by overvoltage events up to $\pm 42\text{ V}$. However, the $50\ \Omega$ input resistor is not protected by this circuit. The resistor can sustain long-term voltages to only 7 V_{rms} . It is therefore important to pay close attention to the maximum signal input when the inputs are set for $50\ \Omega$.

Source impedance

To preserve the high signal quality of the NI 5922, the digitizer must be sourced from a low-impedance source. The source impedance generates thermal noise depending on the value of the impedance. If the impedance exceeds 100 Ω , the thermal noise of the source exceeds the noise performance of the NI 5922. As with any other amplifier, the input bias current is slightly non-linear. Depending on the value of the source impedance, this non-linearity results in a minor non-linear voltage drop created in the source. Because of the very high linearity of the NI 5922, this voltage drop may actually degrade the linearity performance. Linearity performance of the NI 5922 is only guaranteed for source impedances up to 50 Ω .

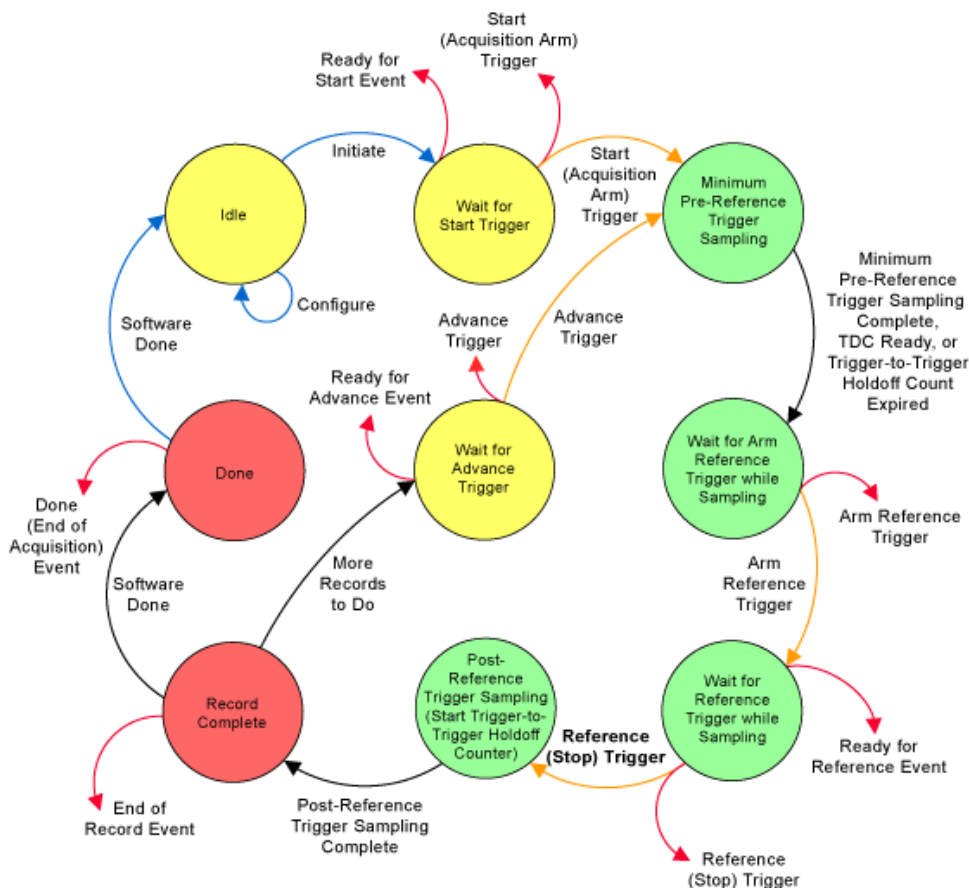
NI 5922 AC/DC/GND Coupling

You can select AC, DC, or GND input coupling for both the 1 M Ω input path and the 50 Ω input path in the NI 5922.

Select AC coupling if the input signal has a DC component that you want to reject, provided that you are not concerned about low-frequency flatness. In AC coupling, the input amplifier is biased from the internal 1 M Ω resistor. Bias current from the input amplifier may create a substantial DC offset. In addition, noise performance of the amplifier decreases at low frequencies because the AC filter source high impedance at low frequencies is high. Ground coupling disconnects the input channel from the signal connected and internally connects the channel to ground to provide a ground reference.

SMC-Based Digitizers Acquisition Engine State Diagram

The following figure shows the acquisition engine state diagram for SMC-Based digitizers.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure a digitizer to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based digitizers can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

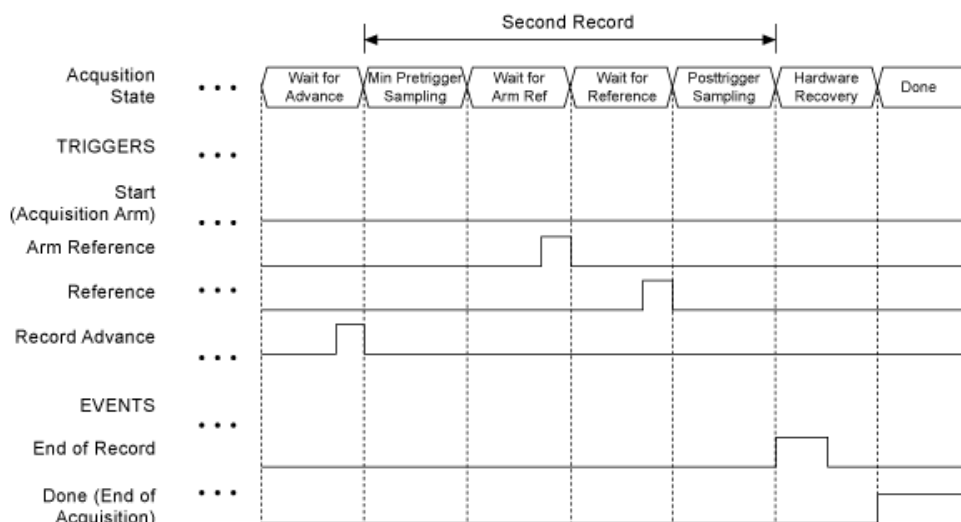
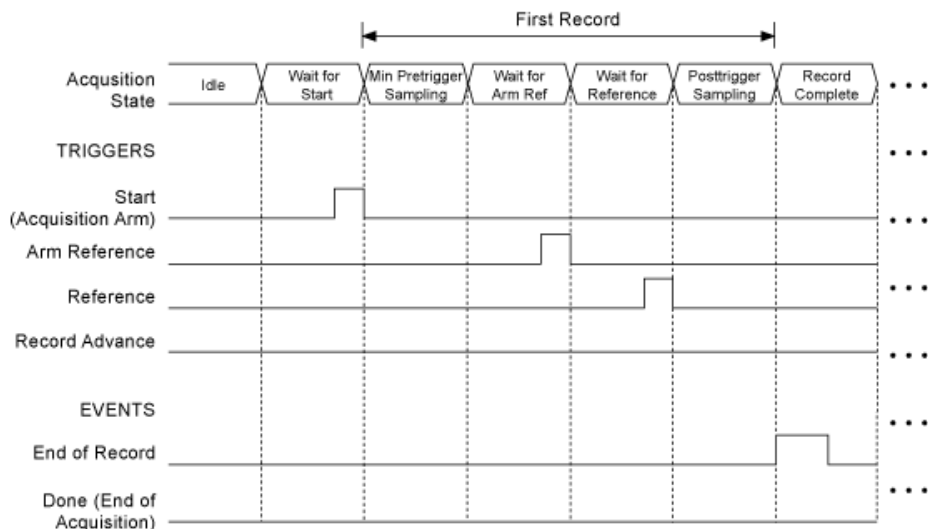
[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)


SMC-Based Digitizers Timing Diagram

SMC-Based digitizers (NI 5105/5114/5122/5124/5142/5152/5153/5154/5160/5162/5622/5922) support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how SMC-based digitizers react to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the digitizer when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.

 Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the digitizer has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

NI PXI-5922 Routing Matrix

The following table shows the signals available for export from the NI PXI-5922 and the lines to which they can be routed.

Signal	Destination		
	RTSI <0..6> (PXI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)
Exported Clocks			
Reference Clock (External)	√	√	√
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

NI PCI-5922 Routing Matrix

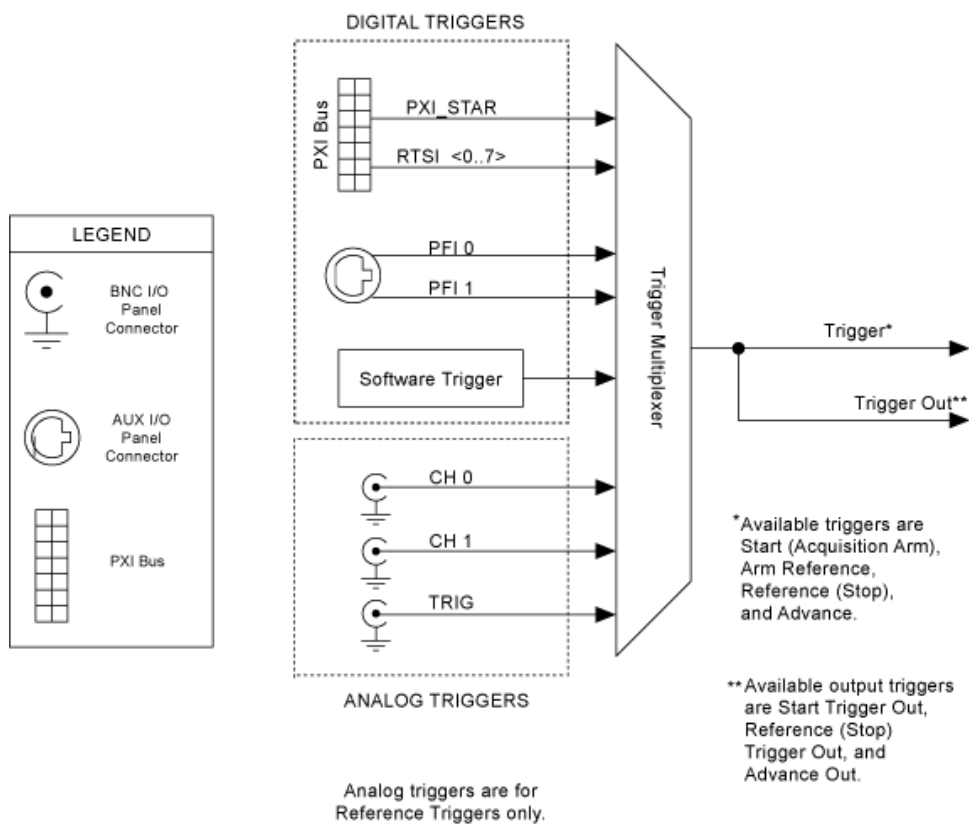
The following table shows the signals available for export from the NI PCI-5922, and the lines to which they can be routed.

Source	Destination			
	RTSI <0..6> (RTSI Bus)	CLK OUT (SMB)	PFI <0..1> (AUX I/O)	RTSI 7

Exported Clocks				
Reference Clock (External)	—	√	√	—
Reference Clock (Internal)	—	√	√	√
Triggers				
Acquisition Arm (Start) Trigger	√	—	√	—
Reference (Stop) Trigger	√	—	√	—
Events				
End of Record Event	√	—	√	—
End of Acquisition Event	√	—	√	—
Ready for Start Event	√	—	√	—
Ready for Reference Event	√	—	√	—
Ready for Advance Event	√	—	√	—

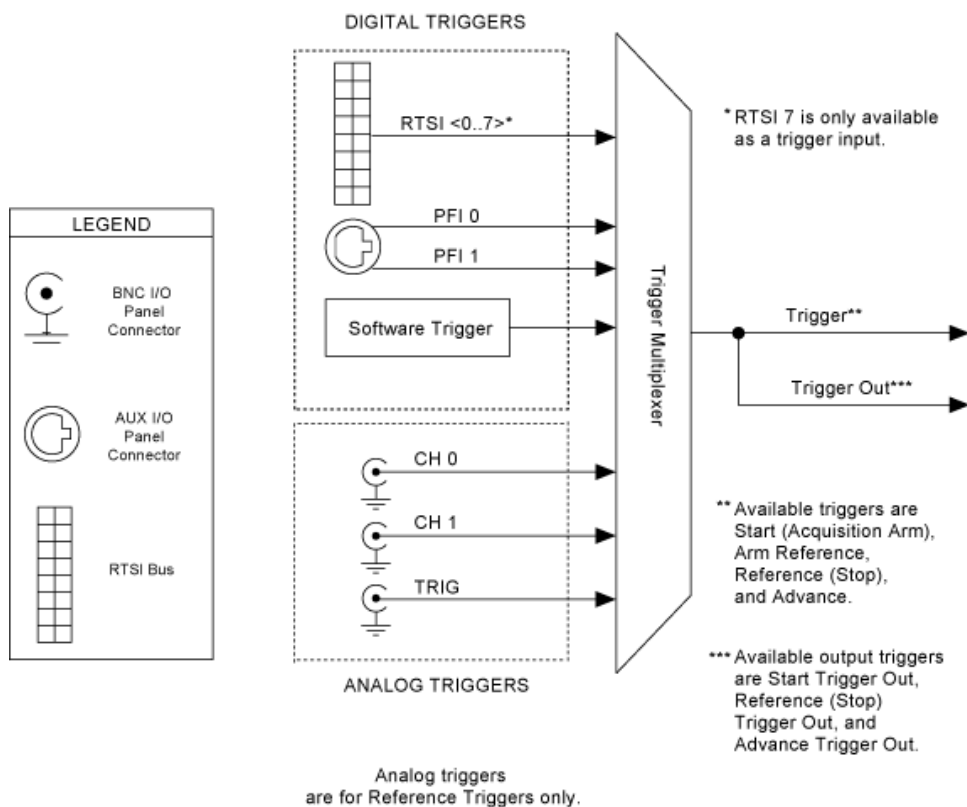
NI PXI-5922 Trigger Sources

The following figure shows the trigger sources for the NI PXI-5922.



NI PCI-5922 Trigger Sources

The following figure shows the trigger sources for the NI PCI-5922.



NI 5922 Trigger Types

The NI 5922 supports the following trigger types: [edge](#), [hysteresis](#), [immediate](#), [digital](#), [software](#), and [window](#) triggers.

The NI 5922 uses the digitized data from the input channel ADC when performing an analog trigger on CH 0 or CH 1. This behavior ensures a level and time trigger accuracy of ± 1 sample. Because of the internal digital filters, there is a 24- to 30-sample delay when you export an analog trigger to another device. The external trigger channel uses analog circuitry to detect trigger events. Therefore, the accuracy is not dependent on the sample rate and the NI 5922 does not have the sample delay when it exports triggers that are received on the external trigger channel.

NI 5922 Trigger Holdoff

On the NI 5922, the [holdoff](#) timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next

record have been acquired, the holdoff timer is evaluated. If the timer has expired, the digitizer arms its Reference Trigger circuit. If the timer has not expired, the digitizer continues pretrigger sampling until the timer expires and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition.

NI 5122/5124/5142/5622/5922 Trigger Delay

Trigger delay, which is specified in seconds, is achieved by adding the appropriate number of posttrigger samples to the record while keeping the allocated onboard memory equal to the record size you request. NI-SCOPE then corrects the trigger time by the delay you specify. To determine the maximum delay for a particular actual sample rate, use the following formula:

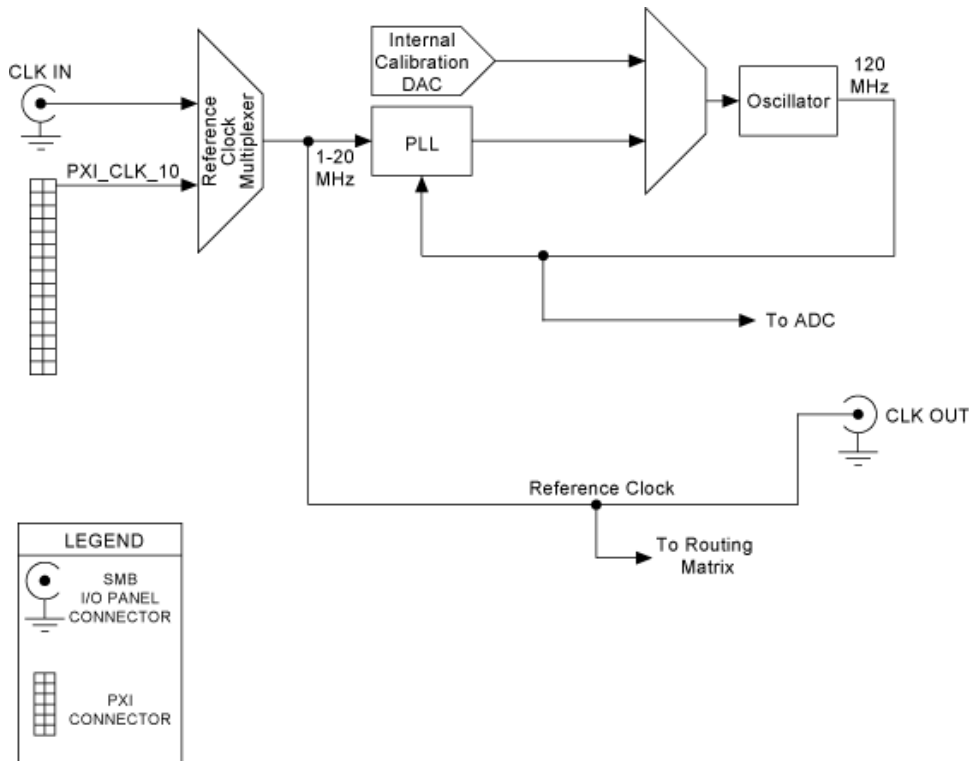
$$\text{Max trigger delay in seconds} = [(2^{34} - 1) - \text{requested posttrigger samples}] \times (1/\text{actual sample rate})$$



Notes The maximum trigger delay changes when you use an external sample clock, and when you sample at rates other than full rate while using the internal sample clock. The maximum trigger delay also changes based on the binary sample width. When binary sample width is set to 8, the maximum trigger delay is doubled. When binary sample width is set to 32, the maximum trigger delay is halved.

NI PXI-5922 Clocking

The clock circuitry on the NI PXI-5922 offers versatile clocking options. You can use the phase-locked loop (PLL) circuit on the NI PXI-5922 to phase lock the internal 120 MHz oscillator with the PXI 10 MHz reference or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PXI-5922.



Sample Clock

The sample clock is sent to the ADC on each channel and to the input timing engine. The NI PXI-5922 has an onboard voltage controlled crystal oscillator (VCXO) running at 120 MHz that is the sample clock on the device. When using the onboard 120 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is generated from the calibrated 120 MHz frequency of the VCXO. In PLL mode, the NI PXI-5922 phase locks its 120 MHz oscillator to the supplied reference clock. The PLL mode is useful when you synchronize the NI PXI-5922 with other devices in a measurement system.

Reference Clock

The reference clock is used in the NI PXI-5922 PLL circuit to synchronize the sample clock to the reference clock. The NI PXI-5922 can accept a reference clock from its front panel (CLK IN) and from PXI_CLK10. This reference clock can be any frequency from 1 MHz to 20 MHz (in 1 MHz increments) if it is provided to CLK IN. The PXI_CLK10 is always a 10 MHz clock. The frequency stability of the sample clock

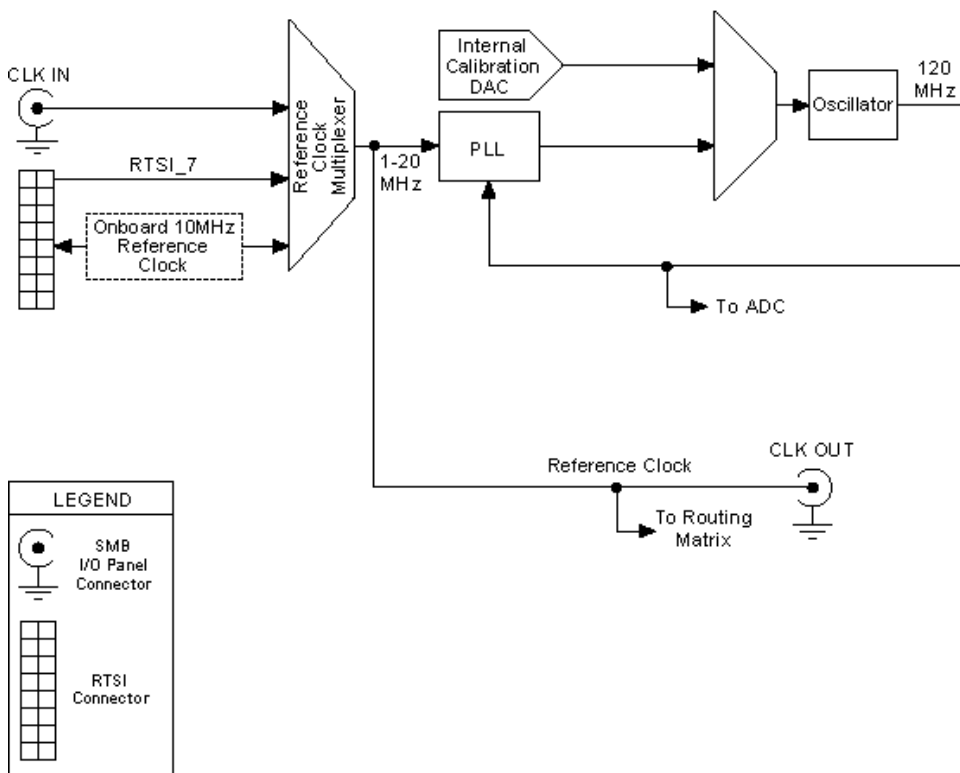
matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PXI-5922 reference clock is None, or not to use a reference clock.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PXI-5922 Routing Matrix](#).

NI PCI-5922 Clocking

The clock circuitry on the NI PCI-5922 offers versatile clocking options. You can use the phase-locked loop (PLL) circuit on the NI PCI-5922 to phase lock the internal 120 MHz oscillator with a reference clock from another module on RTSI 7, or with an external reference clock that you provide. The following diagram shows the clocking options of the NI PCI-5922.



Sample Clock

The sample clock is sent to the ADC on each channel and to the input timing engine. The NI PCI-5922 has an onboard voltage controlled crystal oscillator (VCXO) running at 120 MHz that is the sample clock on the device. When using the onboard 120 MHz oscillator, you can choose either free-run mode or PLL mode. In free-run mode, the sample clock is generated from the calibrated 120 MHz frequency of the VCXO. In PLL mode, the NI PCI-5922 phase locks its 120 MHz oscillator to the supplied reference clock. The PLL mode is useful when you synchronize the NI PCI-5922 with other devices in a measurement system.

Reference Clock

The reference clock is used in the NI PCI-5922 PLL circuit to synchronize the sample clock to the reference clock. The NI PCI-5922 can accept a reference clock from its front panel (CLK IN) as well as from RTSI 7 (RTSI Clock) on the [RTSI bus](#) or the internal 10 MHz reference clock. This reference clock can be any frequency from 1 MHz to 20 MHz if it is provided to CLK IN or to RTSI 7. The internal reference clock is always a 10 MHz clock. The frequency stability of the sample clock matches that of the PLL reference clock when the two are phase locked. In turn, phase locking synchronizes clocks of multiple devices that are phase locked to the same reference clock. The default setting for the NI PCI-5922 reference clock is `None`, or not to use a reference clock.

Exporting Reference Clock

If you are using an external reference clock to phase lock the internal sample clock, you can export the reference clock for use with other instruments. For more information on exporting the reference clock, refer to the [NI PCI-5922 Routing Matrix](#).

NI 5922 Onboard Memory

The NI 5922 allocates at least 640 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus, the minimum size for a buffer in the onboard memory is approximately 64 samples, because each sample is stored as a 32-bit word. Software

allows you to specify buffers of less than these minimum buffer sizes, but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 64 samples. The available memory options are 2 MS, 8 MS, and 64 MS per channel.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. After the trigger is received, the NI 5922 continues to acquire posttrigger samples if you have specified a posttrigger sample count. The acquired samples are placed into onboard memory. The number of posttrigger or pretrigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization

SMC-based digitizers are built on the National Instruments Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

NI 5922 Multiple-Record Acquisition

The NI 5922 supports multiple-record acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the digitizer automatically begins a new acquisition in a new memory record immediately after finishing the previous one. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the digitizer. Between each record, however, there is a dead time during which no triggers are accepted. During this time, the memory controller sets up for the next record. There is also a holdoff from the last trigger in a record to the start of a new record. This means that the actual dead time is the greater of the between-

record dead time and the holdoff time. There may also be additional dead time while the minimum number of pretrigger samples are being acquired. To increase the dead time between records, use the trigger holdoff feature.

The number of records that can be acquired varies depending on the memory option that the device has. Currently the NI-SCOPE software has a limit of 100,000 records that can be acquired without fetching. However, if the records are fetched while they are being acquired, more records can be acquired. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) and [SMC-Based Digitizers Timing Diagram](#) for more information.

NI 5922 Calibration

Every measurement instrument performs within its specifications over some finite temperature range and time period. If the temperature changes and time period exceeds those specified, calibration is required. For example, if the accuracy of a digitizer is specified as $\pm(1\% \text{ of reading} + 10 \text{ mV})$, and you apply 5 V to the input, the error is as follows:

$$1\% \text{ of } 5 \text{ V} + 10 \text{ mV} = 60 \text{ mV for temperature range } 18\text{--}28 \text{ }^\circ\text{C}$$

This example demonstrates the traditional method of specifying accuracy. The problem with the traditional method is that in a system environment, temperature is not easily controlled. When a system is composed of multiple instruments integrated together, the system is subject to temperature rise caused by inherent compromises in air circulation and other factors. Self-heating from surrounding equipment, uncontrolled manufacturing floor environment, and dirty fan filters are among these factors. If the ambient temperature is outside of the 18–28 °C range, you may need to know exactly what the measurement accuracy is to compensate for this temperature variation. With the traditional method, the only way to get the specified accuracy outside of the 18–28 °C range is to externally calibrate the system at the desired temperature. However, an external calibration is time-consuming, expensive, and done infrequently, so the specified accuracy is rarely obtained. You can learn more about external calibration at ni.com/calibration. In the example, if the ambient temperature of the digitizer is 48 °C, assuming the Tempco (TC) error is specified as

$$\text{TC} = (0.1\% \text{ of reading} + 1 \text{ mV}) / \text{ }^\circ\text{C} \text{ (a typical number is } 10\% \text{ of accuracy} / \text{ }^\circ\text{C}),$$

then the additional error is

$$20\text{ }^{\circ}\text{C} \times \text{TC} = \pm(2\% \text{ of reading} + 20\text{ mV}) \text{ or } 120\text{ mV}$$

The total error is three times the specified error (180 mV in the example above, versus 60 mV if temperature effect is ignored) due to the 48 °C ambient temperature.

Self-Calibration

To eliminate errors caused by changing temperatures, NI-SCOPE provides a highly repeatable self-calibration function. For the NI 5922, this self-calibration capability yields the following benefits:

- Corrects for DC gain and offset errors within the digitizer by comparison to a precision, high-stability internal voltage reference. This comparison is done for all ranges and both input impedance paths (50 Ω and 1 MΩ).
- Compensates input bias current.
- Corrects for nonlinearities in the ADCs.
- Takes approximately 2 minutes.

When to Self-Calibrate

For optimum performance, use self-calibration when the digitizer is placed in a new system, any time the temperature changes more than 5 °C from the previous self-calibration, or one week after the previous self-calibration. The result is a product that yields full performance over its operating temperature range and 2-year calibration cycle for DC accuracy, AC response, and trigger level/timing. When the two-year calibration interval expires, an external calibration is required.

The NI 5922 incorporates a temperature sensor that monitors temperature variations. The previous self-calibration time and date can also be read. Unless temperature variations are a serious problem, self-calibration is not recommended more than once per day.

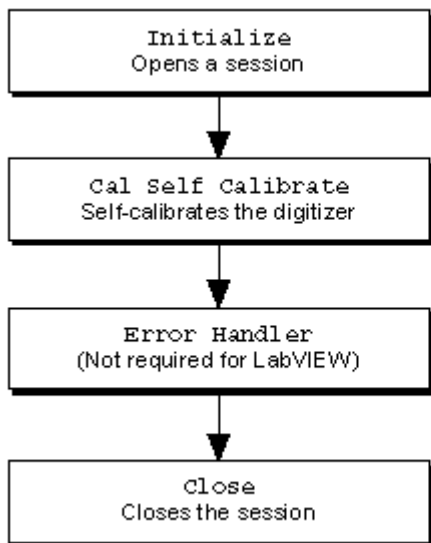
Input Connections During Self-Calibration

The NI 5922 internal circuitry is automatically isolated from the input during self-calibration. However, problems may occur if high-voltage, high-frequency signals (in excess of 500 V/μs slew rate) are present during self-calibration. When in doubt,

disconnect the inputs as directed. If you are absolutely certain that the maximum slew rate of the input signal is below 500 V/μs, then it is acceptable to leave the input signal connected during self-calibration.

Programming Flow

The following diagram shows the typical programming flow for self-calibration.



Summary of Calibration Options

A summary of the calibration options available and when to use them is shown in the following table.

Calibration	Impact	When	Notes
External calibration	Calibrate time drift of onboard reference	Every 2 years	Calibrates and verifies to full specifications
Self-calibration	<ul style="list-style-type: none"> ▪ Offset and gain ▪ Linearity ▪ Bias current 	One week, or when temperature changes >5 °C	Ensures range-to-range matching, trigger accuracy, linearity, and bias current
No calibration	None, within 2 year calibration cycle or if temp	High accuracy not required outside of 5 °C	If self-calibration is not used, performance can not be guaranteed

erature stays within ± 5 $^{\circ}\text{C}$
--

NI Reconfigurable Oscilloscope Devices Help

April 2019

Expand this book to view the topics that describe the following devices in more detail:

- [PXIe-5164](#)
- [PXIe-5170/5171](#)
- [PXIe-5172](#)



Note For the purposes of this documentation, the terms "digitizer" and "oscilloscope" are interchangeable.

To comment on National Instruments documentation, refer to the National Instruments website.

© 2014–2019 National Instruments. All rights reserved.

PXIe-5170/5171

This section includes useful information about the PXIe-5170/5171 reconfigurable oscilloscopes, including front panel connectors, theory of operation, signal path information, block diagrams, and calibration information.



Note NI provides two software options for programming the PXIe-5170/5171: NI-SCOPE instrument driver software and LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes. Refer to the **PXIe-5170/5171 Getting Started Guide** for more information about choosing the right software for your application.

The PXIe-5170/5171 includes the following features:

- Up to 8 channels
- 14-bit resolution with 5 voltage ranges
- 250 MS/s sample rate with internal or external clocking support
- Up to 250 MHz bandwidth (PXIe-5171)

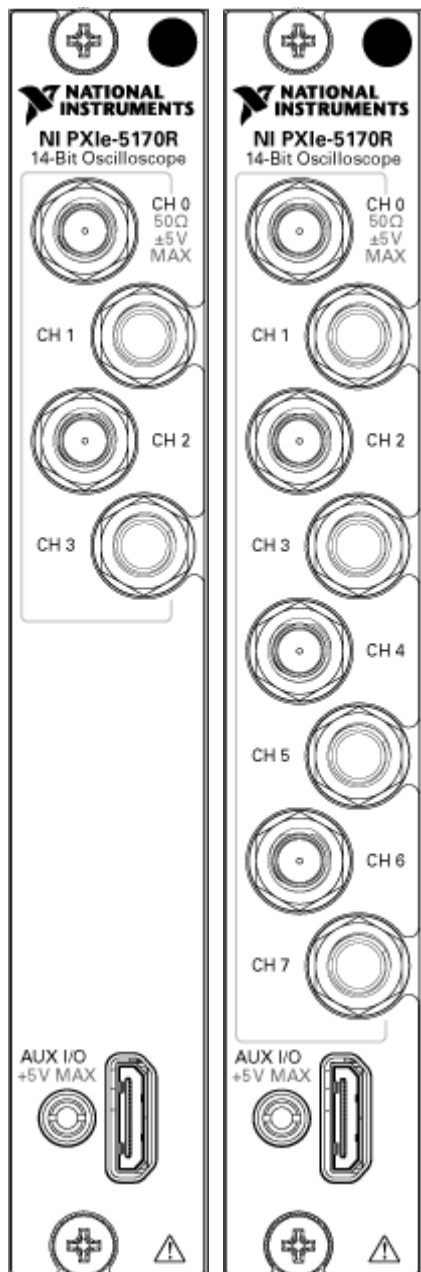
- User-programmable with LabVIEW FPGA Module
- Gen 2 x8 PXI Express
- NI-TClk synchronization using the NI-SCOPE API



Note Refer to the **PXIe-5170 Specifications** or **PXIe-5171 Specifications** documents available at ni.com/manuals for complete hardware specifications.

PXIe-5170 Front Panels

This section describes the front panel and connectors of the 4 CH and 8 CH variants of the PXIe-5170.



Front Panel Connectors

Label	Connector Type	Function
CH 0—CH 3	SMA connector	Analog input terminal
CH 4—CH 7	SMA connector	Additional analog input terminals on the NI 5170R (8 CH)

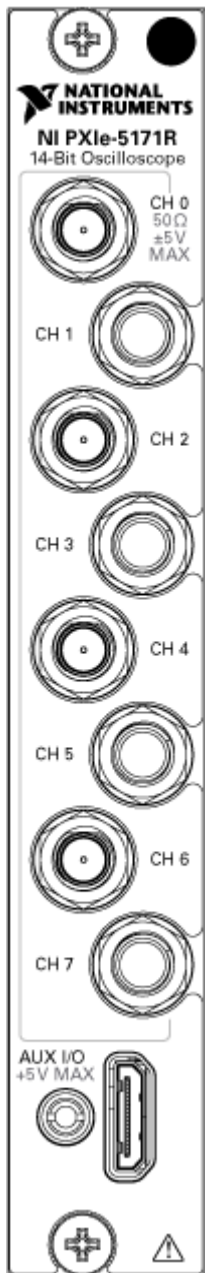
AUX I/O	MHDMR connector	Sample Clock or Reference Clock input, Reference Clock output, bidirectional digital PFI, and 3.3 V power output
---------	-----------------	--



Note The AUX I/O connector accepts a standard, third-party HDMI™ type C cable, but the AUX I/O port is not an HDMI interface and the specified performance of the AUX I/O connector is not guaranteed if a third-party HDMI cable is used. Use NI cable type SHH19-MH19-AUX for all AUX I/O connections. Do not connect the AUX I/O port on the PXIe-5170 to the HDMI port of another device. NI is not liable for any damage resulting from such signal connections.

PXIe-5171 Front Panel

This section describes the front panel and connectors of the PXIe-5171.



Front Panel Connectors

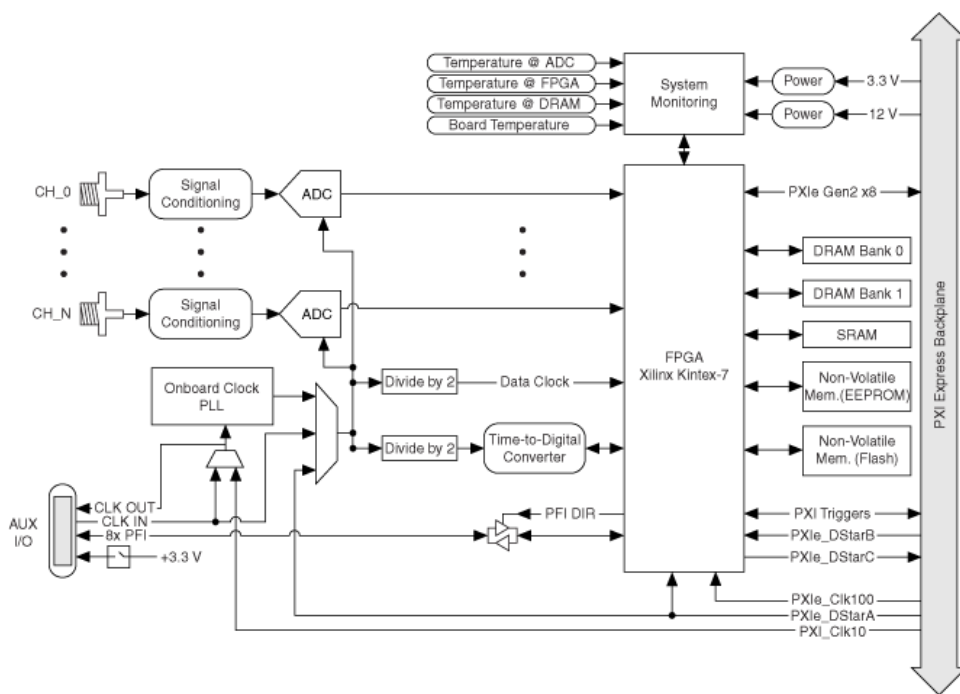
Label	Connector Type	Function
CH 0—CH 7	SMA connector	Analog input terminal
AUX I/O	MHDMR connector	Sample Clock or Reference Clock input, Reference Clock output

, bidirectional digital PFI, and 3.3 V power output

Note The AUX I/O connector accepts a standard, third-party HDMI™ type C cable, but the AUX I/O port is not an HDMI interface and the specified performance of the AUX I/O connector is not guaranteed if a third-party HDMI cable is used. Use NI cable type SHH19-MH19-AUX for all AUX I/O connections. Do not connect the AUX I/O port on the PXIe-5171 to the HDMI port of another device. NI is not liable for any damage resulting from such signal connections.

Block Diagram

The PXIe-5170 and PXIe-5171 are represented by the following high-level block diagram. Channel count has been abstracted.



The PXIe-5170/5171 has the following subsystems:

[Signal Conditioning](#)

[Clocking](#)

[PFI](#)

[PXI Trigger Resources](#)

External Power

DRAM

PXle-5170/5171 System Monitoring and Shutdown

The PXle-5170/5171 has onboard power and temperature monitoring. You can access these measurements through the instrument design libraries and [FPGA I/O](#). These measurements are used as protection limits to prevent device damage in the case of excessive power or temperature across the device.

Thermal Monitoring and Shutdown

The PXle-5170/5171 is protected against excessive temperatures and shuts down in the presence of excessive heat. In the event of an over temperature condition, the system responds at two set points. At an initial limit, the hardware and software will generate an error signal in the FPGA warning the user of a fault condition. Should the fault not be cleared and the temperature reaches a critical limit that may result in device damage, the device will shut itself down.

(NI-SCOPE) If the device powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, such as the commit function and the self-calibration function.
- MAX returns an error message if you run a self-test on your device after it exceeds the thermal shutdown temperature.

Re-enabling the Device After Thermal Shutdown

To re-enable your device after thermal shutdown, use one of the following methods:

- Power down the computer connected to the device.
- Disable and enable the device from the Device Manager.

During normal operation, the thermal protection should not engage over the rated ambient temperature range of the device, provided that the power consumption remains below 100% of the allowed power.

You can read the device temperature using the instrument design libraries. Refer to [Configuration Overview](#) for more information.

Power Monitoring and Shutdown

To stay within PXI Express power and cooling requirements, there may be limitations on the size and activity factor of the logic loaded to the FPGA. You can access onboard power sensors to evaluate the power draw of a particular design, allowing you to balance design capabilities with power limitations.

Should power consumed by the device exceed a factory-programmed limit, the device will generate a signal on an FPGA I/O Node that indicates an over-power condition and shut itself down.

(NI-SCOPE) In the event that the device powers down, you are notified with an error message in one of the following ways:

- NI-SCOPE returns an error when you use any of the functions that program the hardware or check hardware status, such as the commit function and the self-calibration function.
- MAX returns an error message if you run a self-test on your device after it exceeds the power shutdown limit.

Re-enabling the Device After Power Shutdown

To re-enable your device after power shutdown, use one of the following methods:

- Power down the computer connected to the device.
- Disable and enable the device from the Device Manager.


Power Planning

If your device is installed in a chassis and slot that allows for more power than the minimum required by the PXI Express specification, it is possible to operate your device with higher power, as long as the device does not exceed its thermal limits.

This allows you to increase performance if your device is operating in a cool environment. Refer to your chassis documentation to understand how much system and slot power it provides to peripheral modules.

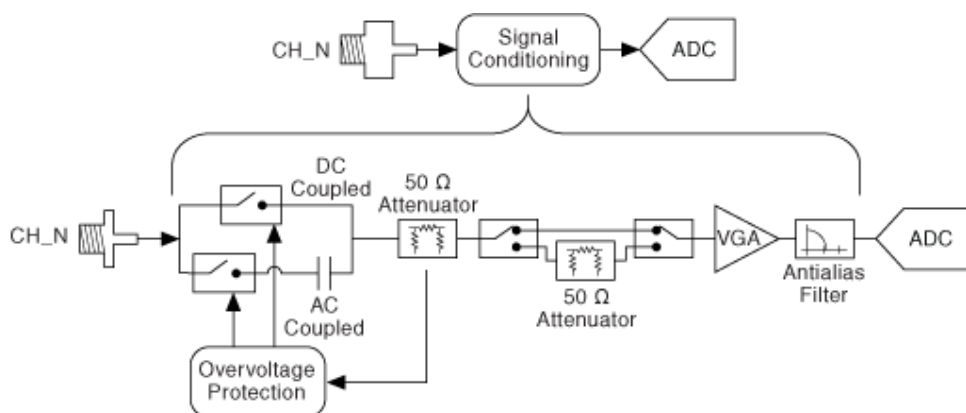
Signal Conditioning

The PXIe-5170/5171 provides users with a range of front end configurations, including selectable ranges, coupling, and filtering.

 Note A variable gain amplifier (VGA) is used in both devices to provide multiple voltage ranges.

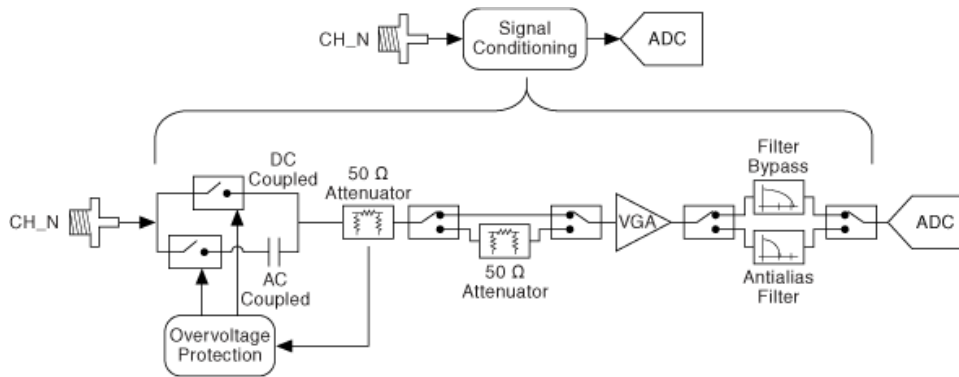
PXIe-5170 Block Diagram

The PXIe-5170 has 4 or 8 channels whose individual high-level functions are represented below.



PXIe-5171 Block Diagram

The PXIe-5171 has 8 channels whose individual high-level functions are represented below.



PXIe-5170/5171 Input Coupling

You can select AC or DC [input coupling](#) with 50 Ω input impedance. AC coupling allows you to reject the DC components of a signal. For information about the AC coupling cutoff frequency or AC flatness, refer to the [device specifications](#).

PXIe-5170/5171 Input Ranges

The PXIe-5170/5171 allows the following software-selectable input voltage ranges (V_{pk-pk}): 0.2 V, 0.4 V, 1 V, 2 V, and 5 V. These ranges are ground referenced, and do not have programmable offset control. Signals exceeding these ranges will be affected by the [protection circuitry](#).

PXIe-5170/5171 Protection

The PXIe-5170/5171 channels are protected for transient and overvoltage conditions. If a signal exceeds the absolute maximum rating of ± 5 V, the protection circuitry will attempt to clamp the signal to a safe voltage level.

If the signal exceeds the input range to the point where damage may occur, the protection circuit will open the two front end relays, disconnecting the signal from the oscilloscope electronics. This event generates a signal to the FPGA which you can monitor for error conditions. After opening, the front end switches will close again every 200 ms. If the overvoltage condition is still present, the switches will re-open. This cycle will continue until the condition is cleared, at which point the switches will close and remain closed.

PXIe-5170/5171 Anti-Alias Filter

The PXIe-5170/5171 has an anti-alias filter path for capturing signals in the first Nyquist zone. This filter is a 100 MHz cutoff filter whose primary purpose is to reject signals that may otherwise alias into the spectrum of the 250 MS/s ADC. For information about the AC flatness of this filter, refer to the [device specifications](#).

The PXIe-5171 allows you to bypass the anti-alias filter to capture higher-frequency, band-limited signals.

PXI Ground Loop Noise

Some electronic systems can induce ground loop noise in measurements taken with instruments with multiple ground connections. Spurious DC and low-frequency content can result. The cause is usually power supply return currents from the motherboard that partially flow through the computer chassis to the power supply housing rather than entirely through the power supply cabling.

Follow these guidelines to mitigate the ground loop noise for PXI devices:

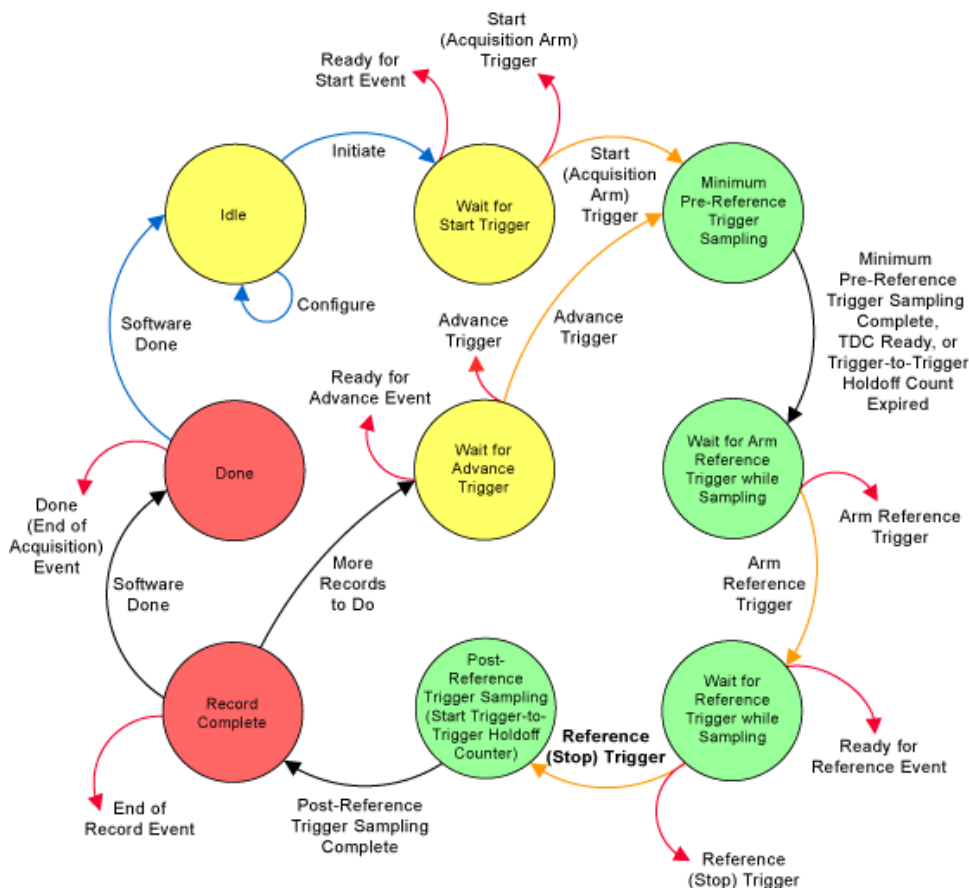
1. Ensure that the screws on the front panel of the device are tightened down.
2. Ensure that any external equipment that is providing a signal, and is not floating with respect to ground (for example, it is not a passive sensor or a battery-operated device), is connected to the same ground as the chassis.
3. Ensure all SMA cables are properly seated and that cable construction or shielding is not compromised.
4. On rack mounted systems, ground the chassis to the metal rack mount to reduce ground loop currents.

SMC-Based Oscilloscopes Acquisition Engine State Diagram



Note This functionality is only available when using NI-SCOPE.

The following figure shows the acquisition engine state diagram for SMC-Based oscilloscopes.



Note The Reference (Stop) Trigger is the same as the trigger level input of any traditional benchtop oscilloscope. To configure the device to behave as a traditional benchtop oscilloscope, configure only this trigger using the niScope Configure Trigger function.

ArrowColor	Indication
Blue	State transitions always caused by software
Black	State transitions caused by the internal state machine of the device
Red	Output signals
Orange	User-configurable state transitions caused by software or hardware

NI SMC-based oscilloscopes can be in any of the following basic states during the course of operation.

Idle—The module is not sampling a waveform. All the session attributes can be programmed in this state. In this state, the attributes have not necessarily been

applied to hardware yet, so the hardware configuration of the module may not match the session attribute values. Also, the module remains configured as it was the last time a session was committed. When `initiate` is called on the module, all the attributes are programmed to the hardware. If the computer has just been reset, or `niScope_ResetDevice` has just been called, the module is in the Idle state.

Wait for Start Trigger—On initiating an acquisition, the module transitions to this state. If the Start (Acquisition Arm) Trigger Source is configured to Immediate, the module immediately transitions out of this state and generates a Start Trigger Event. If the Start Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, it transitions out of this state on the next clock cycle and generates a Start Trigger Event. The default Start Trigger Source is Immediate.

Minimum Pre-Reference Trigger Sampling—The module can transition into this state two ways: receiving the Start (Acquisition Arm) Trigger from the Start (Acquisition Arm) Trigger Source or receiving the Advance Trigger from the Advance Trigger Source. Transitioning into this state depends on the previous state of the module. While in this state, the module samples according to the session attributes configured. The module remains in this state until three conditions are satisfied: the minimum Pre-Reference Trigger sampling completes, the [TDC](#) is ready, and the trigger-to-trigger holdoff count has expired. The minimum Pre-Reference Trigger sampling is at least the user-configured Minimum Record Length multiplied by the user-configured Reference Position. The first time through this state, the trigger-to-trigger holdoff does not have an effect. When the three conditions have been satisfied, the module transitions out of this state on the next clock cycle.

Wait for Arm Reference Trigger while Sampling—After the module finishes the Minimum Pre-Reference Trigger Sampling state, the module transitions into this state. While in this state, the module continues to acquire Pre-Reference Trigger samples according to the session attributes configured. If the Arm Reference Trigger Source is configured to Immediate, the module transitions out of this state on the next clock edge. If the Arm Reference Trigger Source has been configured for a software trigger or a hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module

recognizes a trigger condition, the module transitions out of this state. The default Arm Reference Trigger Source is Immediate.

Wait for Reference Trigger while Sampling—After the module receives Arm Reference Trigger from the Arm Reference Trigger Source, the module transitions into this state. If the Reference Trigger Source has been configured for a software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. When the module recognizes a trigger condition, the module transitions out of this state. The default Reference Trigger Source is Immediate.

Post-Reference Trigger Sampling—After the module receives the Reference (Stop) Trigger, the module transitions into this state. At the beginning of this state, the module starts a trigger-to-trigger holdoff counter. This holdoff counter corresponds to the user-configurable Trigger Holdoff attribute and is used in the Minimum Pre-Reference Trigger Sampling State. You can use the Trigger Holdoff attribute to delay the module from looking for a Reference Trigger between records. At the same time, the trigger-to-trigger holdoff counter is started, the module begins sampling Post-Reference Trigger samples according to the session attributes configured. When the Post-Reference Trigger sampling is completed, the module transitions out of this state.

Record Complete—After the module completes Post-Reference Trigger sampling, the module transitions into this state. The module leaves this state after the current record has been stored in the onboard memory. Upon leaving this state, the module outputs an End of Record Event.

Wait for Advance Trigger—After the module has completed a record and determines that there are still more records to complete, the module transitions into this state. If the Advance Trigger Source is configured to immediate, the module transitions out of this state on the next clock edge. If the Advance Trigger Source has been configured for software or hardware trigger from one of the available sources, the module remains in this state until the configured trigger occurs. Upon the module recognizing a trigger condition, the module transitions out of this state. The default Advance Trigger Source is Immediate.

Done—After the module completes a record and determines that all the records are done, it transitions into this state. Upon entering this state, the module outputs the

End of Acquisition Event, which is a temporary state. The software transitions the module out of this state and back to the Idle state when you call either Fetch or Check Status.

Related Topics

[Triggering](#)

[Trigger Types](#)

[niScope Configure Trigger Poly VI](#)

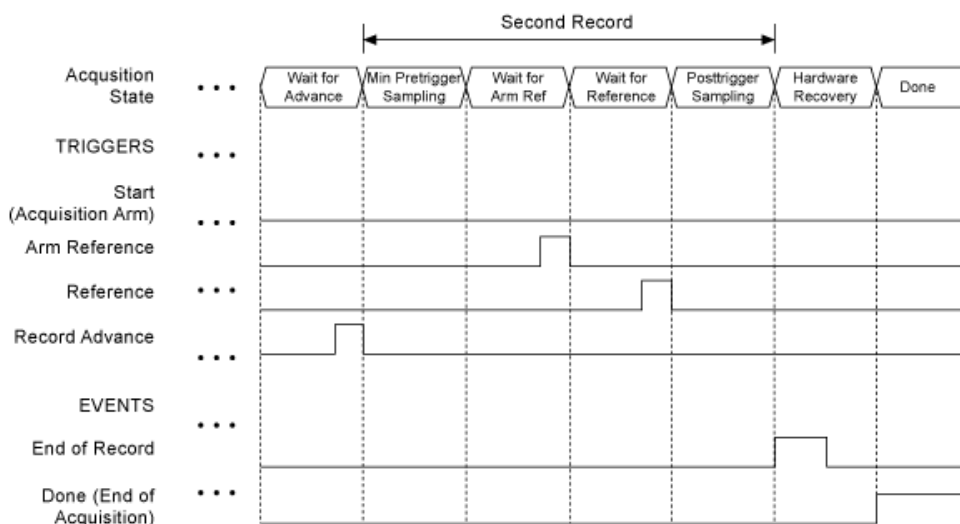
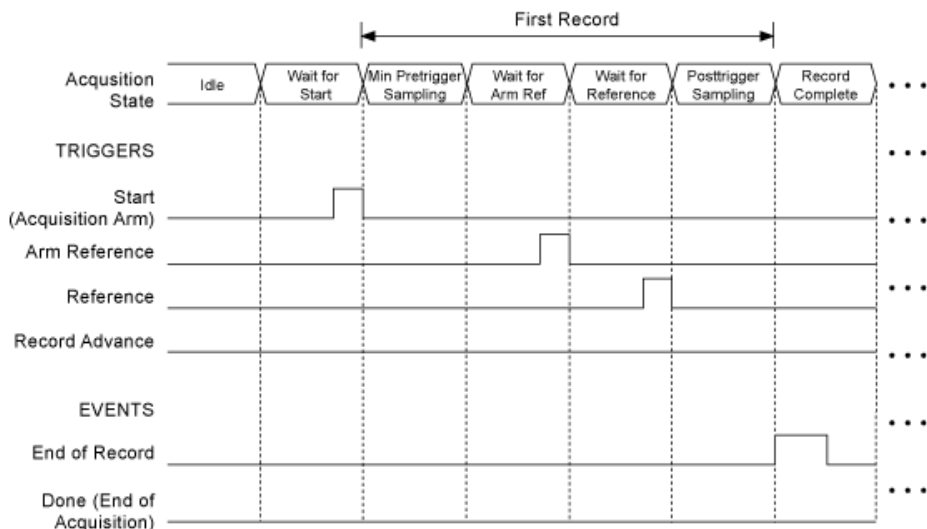
SMC-Based Oscilloscopes Timing Diagram



Note This functionality is only available when using NI-SCOPE.

SMC-Based oscilloscopes support multirecord acquisitions, which allow the capture of multiple triggered waveforms without software intervention. In this mode, the device automatically begins storing a new record to onboard memory a short time after finishing the previous record. The number of records and record size are both configurable.

The following timing diagram illustrates how the PXIe-5170/5171 reacts to the user-configurable input triggers during a multirecord acquisition.



Each state prefixed by **Wait for** is a state in which an input trigger can be configured. This trigger tells the device when to transition out of that particular state. The hardware is only sensitive to a particular trigger when in that trigger's appropriate **Wait for** state. For example, the hardware is not sensitive to a high level on the Advance Trigger until it enters the Wait for the Advance trigger state.



Note The trigger signals in the timing diagram assume active high level triggers.

The exportable events are also shown in the timing diagram. The End of Record Event is generated once per record when the device has acquired all of its pre- and post-Reference Trigger samples. This signal can be used for handshaking between

devices in a system. The Done Event asserts when all of the records have been completed, but it does not assert if the acquisition is aborted or times out.

PXIe-5170/5171 Routing Matrix



Note This functionality is only available when using NI-SCOPE.

The following table shows the signals available for export from the PXIe-5170/5171 and the lines to which they can be routed.

Source	Destination		
	PXI_Trig <0..6>	CLK OUT	PFI <0..1>
Exported Clocks			
Reference Clock	—	√	—
Sample Clock	—	—	—
Triggers			
Acquisition Arm (Start) Trigger	√	—	√
Reference (Stop) Trigger	√	—	√
Events			
End of Record Event	√	—	√
End of Acquisition Event	√	—	√
Ready for Start Event	√	—	√
Ready for Reference Event	√	—	√
Ready for Advance Event	√	—	√

For more information about the routing capabilities of the PXIe-5170/5171, refer to the [PXIe-5170/5171 Block Diagram](#) and [SMC-Based Oscilloscopes Timing Diagram](#).


PXIe-5170/5171 Analog Trigger Types



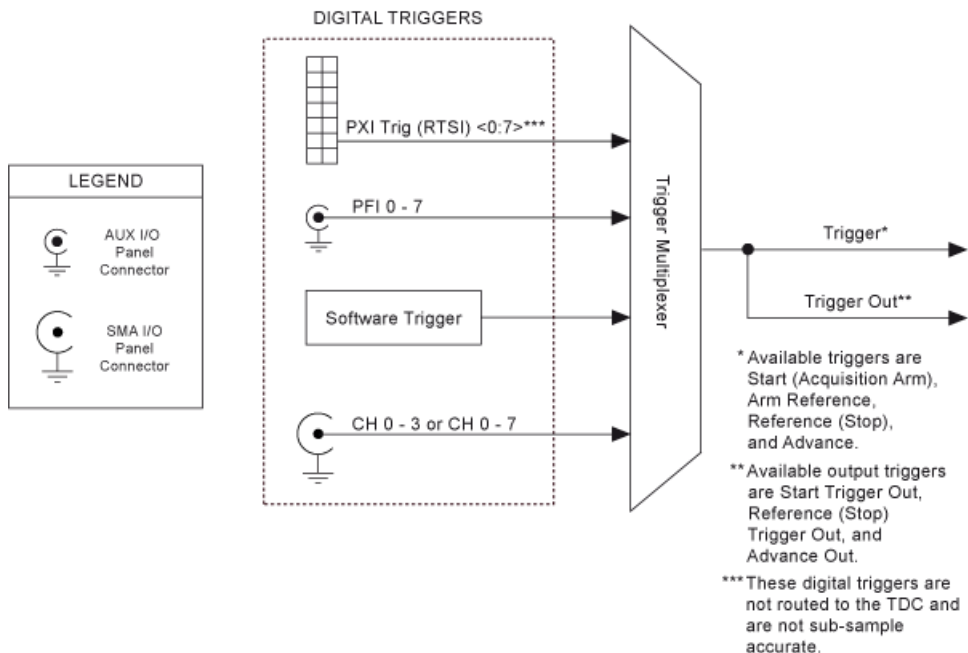
Note By default, this functionality is only available when using NI-SCOPE.

The PXIe-5170/5171 supports analog edge, hysteresis, and window trigger types.

PXIe-5170/5171 Trigger Sources

 **Note** By default, this functionality is only available when using NI-SCOPE.

The following figure shows the trigger sources for PXIe-5170/5171 oscilloscopes.



PXIe-5170/5171 Analog Trigger Paths

The PXIe-5170/5171 provides flexible, high-precision, low-jitter triggering features.

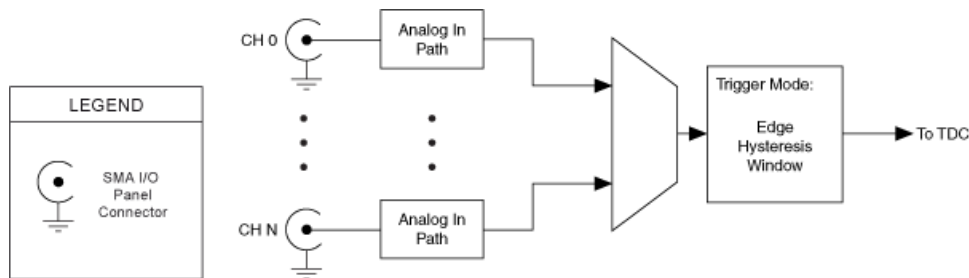
Trigger Channels

You can use a disabled analog input channel as a trigger. For instructions on using a disabled channel to trigger acquisitions, refer to the NI KnowledgeBase.

Analog Trigger Paths

The PXIe-5170/5171 uses the interpolated, predecimated digitized data from the input channel ADC when performing an analog trigger on any of the channels. This

behavior ensures sub-sample level and time trigger accuracy. The following figure shows the analog trigger paths for the PXIe-5170/5171. Channel count has been abstracted.



PXIe-5170/5171 Trigger Holdoff

 Note By default, this functionality is only available when using NI-SCOPE.

For the PXIe-5170/5171, the holdoff timer is started by the Reference Trigger. When the current record finishes and the minimum number of pretrigger samples for the next record have been acquired, the holdoff timer is evaluated. If the timer has expired, the device arms its Reference Trigger circuit. If the timer has not expired, the device continues pretrigger sampling until the timer expires, and then arms its Reference Trigger circuit. Holdoff is applied for each Reference Trigger during a multirecord acquisition. It is meaningless to apply a trigger holdoff on a single-record or continuous acquisition.

For practical purposes, the maximum trigger holdoff time is unlimited.

PXIe-5170/5171 Trigger Delay

 Note By default, this functionality is only available when using NI-SCOPE.

Trigger delay, which is specified in seconds, is achieved by physically delaying the output of the trigger block by a multiple of the 125 MHz Data Clock. NI-SCOPE then offsets the trigger time by an amount of time less than one period of the Data Clock in order to achieve more delay precision than the 125 MHz Data Clock allows on its own.

The maximum trigger delay is 18,014,398 seconds (approximately 6.8 months).

PXIe-5170/5171 FPGA

The PXIe-5170/5171 allows access to the onboard FPGA using LabVIEW FPGA Module, allowing you to directly alter the behavior of the device. This section outlines the digital capabilities of the FPGA and the methods for interfacing between FPGA resources and your application.

Xilinx Kintex-7 FPGA

The PXIe-5170 contains a Xilinx Kintex-7 XC7K325T FPGA, and the PXIe-5171 contains a Xilinx Kintex-7 XC7K410T FPGA.

The FPGA is used for HOST interaction over the PCI Express interface, system configuration, digital data movement, and digital signal processing. The FPGA has direct connections to:

- device ADCs
- the PCI Express bus (through DMA FIFOs, Controls, and Indicators)
- DRAM
- PFI
- Synchronization resources
- PXI trigger lines

These connections allow for custom programming using LabVIEW FPGA Module to meet the needs of many types of applications.

Reconfigurable FPGA Resources

The FPGAs in the PXIe-5170/5171 have the following resources:

Resource Type	PXIe-5170	PXIe-5171
Slice registers	407,600	508,400
Slice look-up tables (LUT)	203,800	254,200
DSPs	840	1,540
18 Kb block RAMs	890	1,590

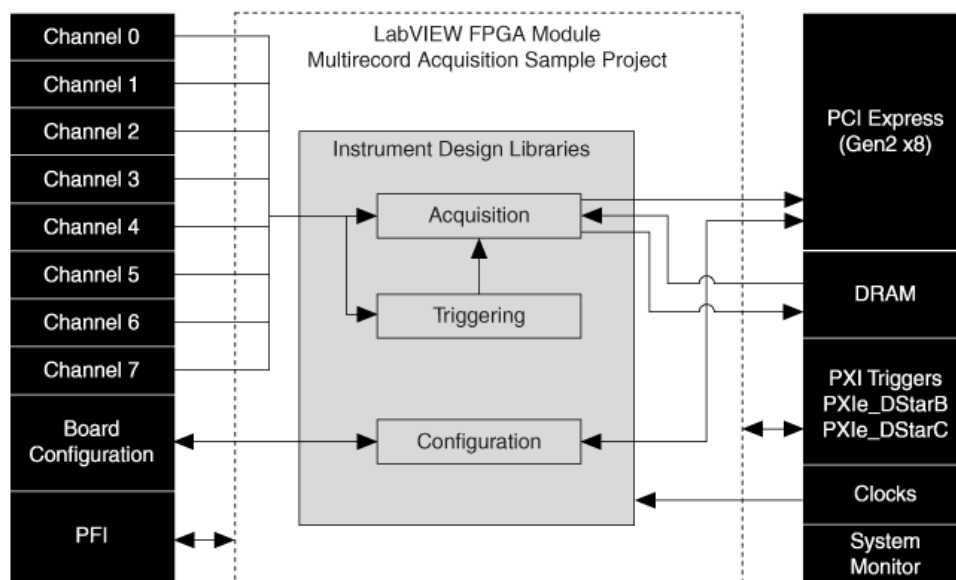
Note that some of these resources are consumed by the logic necessary to operate the device and integrate with LabVIEW FPGA Module, and are thus out of the control of users. For information on determining how many resources your design utilizes or requires, refer to the [LabVIEW FPGA Module Help](#).

FPGA I/O Elements



Note This functionality is only available when using instrument design libraries.

Portions of the FPGA are reserved for fixed functions of your device and are not intended for customization. Other logic is provided to users as LabVIEW FPGA source and can be used as-is or edited to customize the instrument for specific applications. For example, the PXIe-5170/5171 can be partitioned, as illustrated below, when you use the [Simple Acquisition \(Multirecord\) Sample Project](#).



Black blocks contain fixed HDL implementations developed by National Instruments and exposed through LabVIEW FPGA I/O Nodes or Methods.

Grey blocks contain modular LabVIEW FPGA IP distributed with LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes (instrument design libraries). You may use all or some of the instrument design libraries to customize your device. The instrument design libraries are most effective when you use them to build custom

logic, rather than modifying the code inside the libraries themselves. You can interface with these blocks on the LabVIEW block diagram.

The white, dotted-line block contains LabVIEW FPGA code made available through LabVIEW Sample Projects. The sample projects are installed with LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes. Sample projects give you a ready-to-run application built using instrument design libraries, and are a good starting point for customizing your device behavior. You can also add third party IP, HDL, or other custom logic to further customize your device.

These blocks interface with each other using LabVIEW FPGA I/O elements, such as I/O Nodes and I/O Methods.

For more information on implementing custom logic, adding third-party IP, and using I/O Nodes and I/O Methods, refer to the [LabVIEW FPGA Module Help](#).

Clocks



Note This functionality is only available when using instrument design libraries.

The following table lists the clocks available in the FPGA. In addition to these clocks, LabVIEW FPGA allows for derived clocks at user-defined frequencies, using these clocks as references. Not all clock sources may be used to derive clocks. For more information on how to create a derived clock, refer to the Creating FPGA-Derived Clocks topic in the [LabVIEW FPGA Module Help](#).



Note The clocks in this table are available only when programming the FPGA target using LabVIEW FPGA and the instrument design libraries.

Name	Frequency (MHz)	Supports Derived Clocks?	Description
40 MHz Onboard Clock	40	Yes	Free-running 40 MHz oscillator.
PX1e_Clk100	100	Yes	100 MHz clock from the backplane.
Data Clock	125	No	ADC Clock (Sample Clock) divided in hardware by 2, provided to the ADC interface and LabVIEW FPGA, and used by t

			he instrument design libraries. For more information on Sample Clock sources, refer to the Clocking section.
Data Clock x2	250	No	Data clock multiplied by two. This clock is useful for running overclocked DSP. For more information on Sample Clock sources, refer to the Clocking section.
Data Clock x3	375	No	Data clock multiplied by three. This clock is useful for running overclocked DSP. For more information on Sample Clock sources, refer to the Clocking section.
PXIe_DStarA	120 to 250	Yes	External backplane clock driven by the System Timing Module.

I/O Resources



Note This functionality is only available when using instrument design libraries.

Use FPGA I/O nodes to connect to the device I/O. You can configure FPGA I/O nodes for reading or writing. The FPGA I/O nodes connect to the front panel I/O, backplane I/O, and internal status signals of the module. The following tables describe LabVIEW FPGA I/O resources visible in the LabVIEW project.



Note Each resource must be used inside of its Required Clock Domain. For more information about clock domains in LabVIEW FPGA Module, refer to the Implementing Multiple Clock Domains topic in the [LabVIEW FPGA Module Help](#).

Sample Data In

Name	Description	Data Type	Access Method	Required Clock Domain
------	-------------	-----------	---------------	-----------------------

Channel N	Contains raw data from the channel N ADC, where N is a channel number from 0 to the maximum channel number of your device. The data is presented as 2 samples per cycle at 125 MHz loop rate.	Array of two fixed-point numbers ¹ $\pm 14, 1$ ²	Read I/O Node	Data Clock
Channel Data Valid	For each channel, this array contains TRUE if the channel's sample is valid and FALSE if the channel's sample is not valid.	Array of four or eight Boolean values, depending on device channel count	Read I/O Node	Data Clock
Channel Overload	For each channel, this array contains TRUE if an overvoltage condition occurred on the channel and FALSE if no overvoltage condition occurred on the channel.	Array of four or eight Boolean values, depending on device channel count	Read I/O Node	Any

¹ The sample at index 0 corresponds to the first sample read from the ADC, while the sample at index 1 corresponds to the second sample read from the ADC.

² The $\pm 14, 1$ notation corresponds to a signed, fixed-point number with a 14-bit word length and a 1-bit integer word length.

Time-to-Digital Converter (TDC)

Name	Description	Data Type	Access Method	Required Clock Domain
TDC Enable	Setting this value to TRUE enables the Time-to-Digital Conversion circuit (Boolean	Write I/O Node	Any

	TDC). Setting this value to FALSE disables the TDC.			
TDC Assert	Setting this value to TRUE asserts a digital signal (a "TDC pulse") synchronously the Data Clock signal.	Boolean	Write I/O Node	Data Clock
TDC Deassert	Setting this value to TRUE asserts a digital signal (a "TDC pulse") synchronously to the PXIe Clk100 signal.	Boolean	Write I/O Node	PXIe_Clk100
TDC Expanded Pulse	Signal from the TDC that is an expanded pulse (500x) generated from the use of TDC Assert and TDC Deassert. The pulsewidth of this signal is proportional to the delay between edges of Data Clock and PXIe Clk100 .	Boolean	Read I/O Node	Any

Trigger and Digital

Name	Description	Data Type	Access Method	Required Clock Domain
PFI N	Controls PFI line N , where N is a PFI line number from 0 to 7. This signal is connected with the AUX I/O front panel connector of the module. To access this digital I/O	Boolean	Read I/O Node, Write I/O Node, I/O Method	Any

	line, use an FPGA I/O node configured for reading or writing, or use the Set Output Enable and Set Output Data methods.		
--	---	--	--

PXI

Name	Description	Data Type	Access Method	Required Clock Domain
PXI_TrigN	Controls PXI trigger line N, where N is a PXI trigger line number from 0 to 7. To access this digital I/O line, use an FPGA I/O node configured for reading or writing along with the Set Output Enable and Set Output Data methods. For proper device and PXI system functionality when using PXI triggers with the LabVIEW FPGA Module, follow the guidelines in PXI Triggers .	Boolean	Read I/O Node, Write I/O Node, I/O Method	Any
PXIe_Sync100	This signal becomes TRUE one PXIe_CLK100 cycle before each rising edge of PXI_CLK10. It can be sampled and used as an enable signal to mirror th	Boolean	Read I/O Node	PXIe_Clk100

	e behavior of PXI_CLK10.			
PXIe_DStarB	Trace length matched signal driven from the System Timing Module slot to the PXIe-5170/5171.	Boolean	Read I/O Node	Any
PXIe_DStarC	Trace length matched signal driven from the PXIe-5170/5171 to the System timing module.	Boolean	Write I/O Node	Any

Device Status

You can monitor device status, power, and temperature using instrument design libraries. For more information on device monitoring, refer to the overview of the [Configuration](#) instrument design library.

Name	Description	Data Type	Access Method	Required Clock Domain
Module Temperature	This resource provides the FPGA temperature reading from the on-chip temperature sensor. To calculate the value in degrees Celsius, use the following transfer function: Temperature (°C) = [(Temperature Measurement * 503.975) / 4096] - 273.15	16-bit signed integer	Read I/O Node	40 MHz Onboard Clock
Module Power Consumption	This resource provides the current device power consumption reading. T	16-bit signed integer	Read I/O Node	40 MHz Onboard Clock

	<p>his instantaneous power reading is the sum of the power consumption reports of the 3.3 V sensor and the 12 V sensor. The reading is in centiwatts (cW). To calculate the value in Watts, use the following transfer function:</p> <p>Power Consumption (W) = Power Measurement / 100</p>			
Temperature Error	<p>Returns TRUE when the module has exceeded its safe operating temperature range. If this condition occurs, shut down the chassis and check for proper cooling.</p>	Boolean	Read I/O Node	Any
AUX I/O Power Enable	<p>Setting this value to TRUE enables the 3.3 V power source from the AUX I/O front panel connector.</p>	Boolean	Write I/O Node	Any
AUX I/O Power Fault	<p>Returns TRUE if an overcurrent condition has occurred at the 3.3 V power source from the AUX I/O front panel connector. For more information, refer to External Power.</p>	Boolean	Read I/O Node	Any

I/O Methods



Note This functionality is only available when using instrument design libraries.

Name	Description	Clock Domain
Device Registers	Function used to read and write registers in the fixed portions of the design. These methods are intended for use only by the instrument design libraries. Use of this method outside of the instrument design libraries may result in aberrant behavior.	40 MHz Onboard Clock

PFI Trigger I/O Methods

Name	Description	Clock Domain
Set Output Data	Writes a boolean to the digital line without enabling the line.	Any
Set Output Enable	Determines whether the digital line reads external data or writes output. Wiring TRUE to Set Output Enable allows the digital line to write data. Wiring FALSE to Set Output Enable allows the digital line to read external data.	Any

PXI Triggers



Note This functionality is only available when using instrument design libraries.

You can use the FPGA I/O Node to access the eight PXI trigger lines, eight PFI lines, and two DSTAR differential trigger lines on the PXIe-5170/5171. The Data Trigger instrument design library contains IP that can operate on signals received on any of these trigger resources.

PXI Trigger lines are bussed across the backplane to all peripheral devices, so they must be reserved by the application to prevent double-driving the lines and potentially damaging third-party PXI Express devices.

Since the eight PXI trigger lines are bidirectional signals, their configuration must include a direction control, which is accessed through an [I/O Method](#). Alternately, PXIe_DStar_B is a dedicated point-to-point signal driven **from** the System Timing Module slot while PXIe_DStar_C is a dedicated point-to-point signal driven **to** the System Timing Module slot.

When developing an FPGA VI that uses these trigger resources, [reserve the trigger lines](#) you are using to ensure compatibility with other NI devices. Refer to the PXI Specifications Tutorial for more information about trigger bus requirements.

Reserving PXI Trigger Resources



Note This functionality is only available when using instrument design libraries.

National Instruments recommends that you reserve the trigger lines used by NI PXI Express devices. If two PXI Express devices try to drive the same trigger line in different applications or if the PXI Express devices are not programmed to work together, the application does not work and, in some cases, third-party PXI Express devices can be damaged. You can use Measurement & Automation Explorer (MAX) or the LabVIEW FPGA host VI to reserve trigger lines.

Reserving Trigger Lines in MAX

If you download and run the FPGA VI interactively, you should configure the PXI Express triggers in MAX. MAX maintains the trigger reservation for the NI device even after you cycle power to the PXI Express chassis.

Reserving Trigger Lines in the LabVIEW FPGA Host VI

If you download and run the FPGA VI programmatically, you should reserve the PXI Express triggers in the host VI. Use the Invoke Method function to reserve the trigger or release the trigger reservation. LabVIEW releases the trigger reservation for the NI device automatically when you close the FPGA VI reference. You must run the host VI again to reserve the trigger.

Complete the following steps to reserve a trigger line for an NI PXI Express device using the host VI.

1. Place the Open FPGA VI Reference function on the block diagram and configure it for the FPGA device and FPGA VI.
2. Place the Invoke Method function on the block diagram.
3. Wire the FPGA VI Reference Out output of the Open FPGA VI Reference function to the FPGA VI Reference In input of the Invoke Method function.
4. Wire the error out output of the FPGA VI Reference function to the error in input of the Invoke Method function.
5. Click the Invoke Method function and select Reserve PXI Trigger from the shortcut menu.
6. Right-click the Trigger input and select Create»Constant. An enum constant is created to help you select the trigger.

To reserve multiple trigger lines, repeat steps 2 to 6 for each trigger line you want to reserve, wiring the FPGA VI Reference Out output of the existing Invoke Method function to the FPGA VI Reference In input of the Invoke Method node that follows it.

Releasing Trigger Lines

Complete the following steps to release a trigger line for an NI PXI Express device.

1. Place the Open FPGA VI Reference function on the block diagram and configure it for the FPGA device and FPGA VI.
2. Place the Invoke Method function on the block diagram.
3. Wire the FPGA VI Reference Out output of the Open FPGA VI Reference function to the FPGA VI Reference In input of the Invoke Method function.
4. Wire the error out output of the FPGA VI Reference function to the error in input of the Invoke Method function.
5. Click the Invoke Method function and select Unreserve PXI Trigger from the shortcut menu.

- Right-click the Trigger input and select **Create»Constant**. An enum constant is created to help you select the trigger.

To release multiple trigger lines, repeat steps 2 to 6 for each trigger line you want to release, wiring the FPGA VI Reference Out output of the existing Invoke Method function to the FPGA VI Reference In input of the Invoke Method node that follows it.

Configuring Trigger Pulses



Note This functionality is only available when using instrument design libraries.

To ensure compatibility with other NI devices, configure trigger pulses on an NI PXI Express device to last for at least two clock cycles of the clock on the receiving device. For example, if the clock on the receiving device is 8 MHz, which is a clock period of 125 nanoseconds, the trigger line must be constant for at least 250 nanoseconds, which is two cycles of an 8 MHz clock.



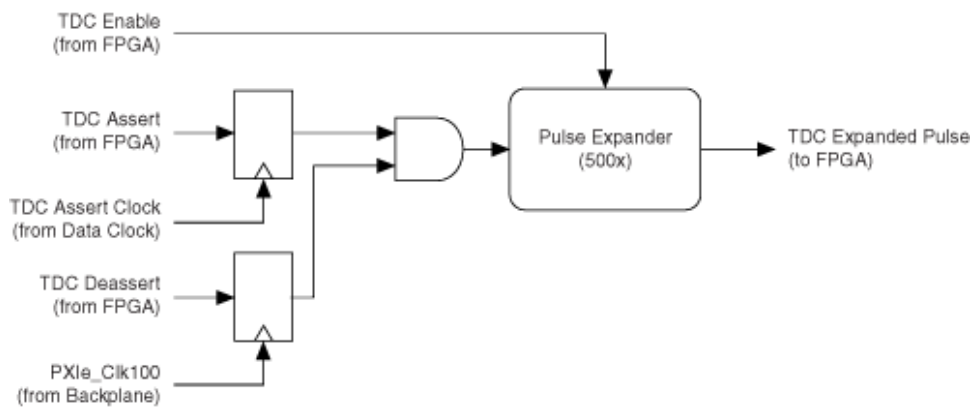
Note Regardless of the clock speed, pulses on the trigger line must be constant for at least 18 nanoseconds. Refer to the PXI Specifications Tutorial for more information about trigger timing parameters.

The clocks between an NI PXI Express device and another device might not be perfectly synchronized. If you assert a trigger line on a NI PXI Express device, you cannot determine at what point in the clock period the trigger registers in the receiving flip-flop. If the trigger arrives during the setup or hold time of the receiving flip-flop, you cannot determine the state of the line for that clock period. Asserting the trigger pulse for two clock cycles ensures that at least one clock cycle on the receiving flip-flop registers as a rising edge and transfers as a trigger.

Time-to-Digital Converter

The PXIe-5170/5171 has a Time-to-Digital Converter (TDC) subsystem that allows the device to measure time between an event asserted in the [Sample Clock](#) domain and an event asserted on the PXIe_Clk100 domain. This measurement may be useful for evaluating the relative timing between synchronous events and is a useful

component for multi-board synchronization. A functional illustration of the TDC is shown below.



(Instrument Design Libraries) To access TDC signals, use [I/O Nodes](#).

PXle-5170/5171 DRAM



Note This functionality is only available when using instrument design libraries.

The PXle-5170/5171 has two banks of dynamic random-access memory (DRAM), which are independently accessible from the FPGA. Refer to the [device specifications](#) for information about the DRAM size and throughput. These DRAM banks are general-purpose, but they are often used for storing acquired waveforms.

Configuring DRAM with FPGA Memory Items



Note This functionality is only available when using instrument design libraries.

Use the FPGA memory item interface to use DRAM in the same way that you use block memory and look-up tables (LUT). DRAM memory items appear in the Project Explorer window under the FPGA target. The FPGA memory item interface allows you to partition the physical DRAM banks into multiple memory items.

Complete the following steps to configure DRAM with FPGA memory items.

1. To create a target-scoped memory item, right-click the FPGA target in the Project Explorer window and select **New»Memory** from the shortcut menu. The Memory Properties dialog box appears.

2. Configure the memory item in the Memory Properties dialog box. Click OK.
The memory item is now populated in the Project Explorer window under the target.
3. Use the memory item in an FPGA VI.

PXle-5170/5171 External Power



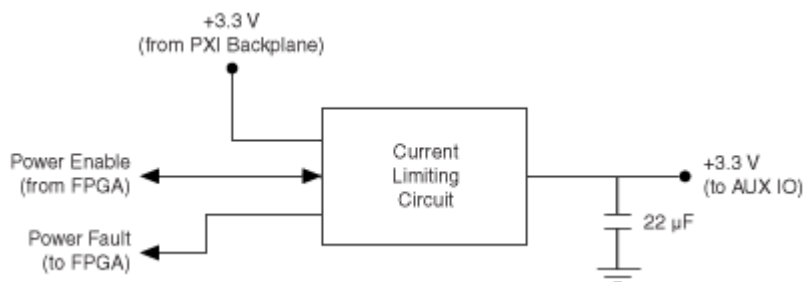
Note This functionality is only available when using instrument design libraries.

The PXle-5170/5171 is capable of providing 3.3 V of power (200 mA nominal) to external devices through the AUX I/O connector. Power drawn from this resource is limited by a current-limiting controller. If a short or partial short occurs during operation while external power is enabled, current sensors will disable the output momentarily. After 300 ms (nominal), the power will be reconnected and checked for an overcurrent condition again. This cycle occurs every 300 ms (nominal) until the fault is cleared.



Note Behavior of this circuit is affected by component temperature. Actual protection limits may change with device and ambient temperature.

By default, power is disabled. To enable and monitor external power, use the Device Status I/O resources. For more information about these I/O resources, refer to [I/O Resources](#).



For the maximum current of the device, refer to the [device specifications](#).

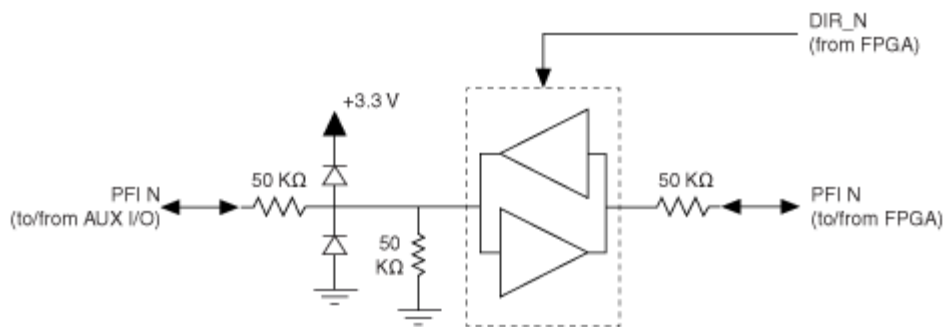
PFI

The PXle-5170/5171 includes eight general purpose digital lines. You can connect to these resources through the front panel AUX I/O port using a supported cable and accessory, such as the SCB-19 (NI part number 156053A-01). These signals are 3.3 V

LVC MOS bidirectional digital signals and may be used in a variety of applications. Direction is controlled independently for each channel through the LabVIEW FPGA Module diagram.

The PFI signals are connected to the FPGA through 3.3 V LVC MOS buffers. These buffers allow for direction control, isolation to protect the FPGA from overvoltage conditions, and excellent signal quality through the matched 50 Ω output impedance. For exact I/O levels and input and output impedances, refer to the [device specifications](#).

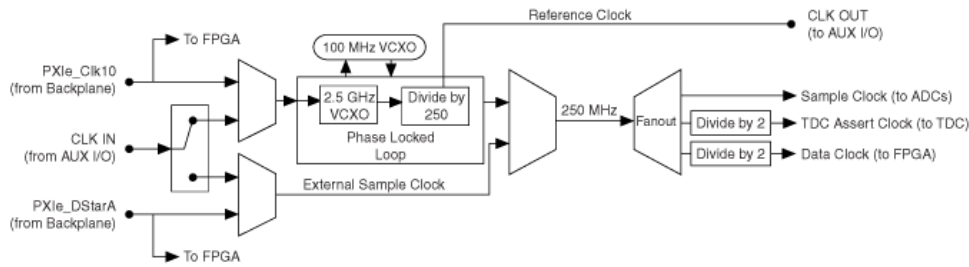
The digital lines are protected against overvoltage conditions. The device provides this protection through a combination of diode clamps to the +3.3 V and GND lines and a positive temperature coefficient resistor for impedance matching.



Note The interface from the PFI LVC MOS buffer to and from the FPGA is bidirectional. To guarantee that this line is not double-driven by both the FPGA and the buffer at the same time, the FPGA implements a direction control latency. This latency is an explicit delay between enabling the FPGA I/O buffer and setting the direction of the PFI. For more information on direction control latency, refer to the [device specifications](#).

PXIe-5170/5171 Clocking

The PXIe-5170/5171 has flexible clocking circuitry, which allows you to generate a 250 MHz clock locked to an internal or external Reference Clock or route an external clock as a source. The main clock sourced from this circuitry is the Sample Clock, which is used to clock the ADCs and the related FPGA and synchronization logic. To configure device clocks from the host, use the [Configuration](#) instrument design library.



Sample Clock

The Sample Clock may either be generated from a 250 MHz phase-lock loop (PLL) locked to a user-selectable Reference Clock or from a user-provided external clock source connected to the AUX I/O CLK IN connector or to the PXIe_DStarA backplane signal. In the FPGA, the Data Clock is a clock source which is derived from the Sample Clock. The Data Clock runs at half the frequency of the Sample Clock. Because of this, two ADC samples per channel are provided to the FPGA for each cycle of the Data Clock.

This clock is routed to the ADCs, the FPGA, and the Time-to-Digital Conversion (TDC) circuits.



Note When using an external Sample Clock, the performance of the instrument will be dependent on the quality of that clock source. For the requirements of the CLK IN terminal, refer to the [device specifications](#).

Reference Clock

The Sample Clock may be phase locked to different reference sources. You may lock the PLL to an internal reference, to the PXIe_Clk10 signal from the backplane, or to an external reference that you provide on the AUX I/O CLK IN pin.



Note When using an external Reference Clock, the performance of the instrument will be dependent on the quality of that clock source. For the requirements of the CLK IN terminal, refer to the [device specifications](#).

Exported Clock

You can use the PXIe-5170/5171 AUX I/O front panel connector to export the Reference Clock through the CLK OUT terminal. This signal will be exported using a 3.3 V LVCMOS buffer.

SMC-Based Oscilloscopes Multiple-Record Acquisition



Note This functionality is only available when using NI-SCOPE.

SMC-based oscilloscopes support multiple-record acquisition, which allows the capture of multiple triggered waveforms without software intervention. In this mode, the oscilloscope automatically begins a new acquisition in a new memory record soon after finishing the previous record. Multiple-record acquisitions can quickly acquire numerous triggered waveforms because they allow hardware rearming of the oscilloscope. Between each record, there is a dead time during which no triggers are accepted. During this time, the device sets up for the next record, as it transitions through the subsequent states of the SMC-Based Oscilloscopes Acquisition Engine State Diagram. There is also a holdoff between the last trigger in a record and the trigger of a new record. This means that the minimum time between triggers is the greater of either:

- The between-record dead time plus the time per record, or
- The user-specified holdoff time (by default, the holdoff time = 0 s).

To increase the minimum time between triggers, use the trigger holdoff feature. For more information, refer to the [SMC-Based Oscilloscopes Acquisition Engine State Diagram](#).

Some oscilloscopes specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample).

The number of records that can be acquired varies depending on the memory option of the device. Depending on the device, NI-SCOPE limits to approximately 100,000 records that can be configured without fetching during the acquisition. However, if an application allows for fetching records while they are being acquired, NI-SCOPE allows more records to be configured. Refer to [Acquiring More Records Than Fit in Digitizer Memory](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

[Making Multiple-Record Acquisitions](#)

PXIe-5170/5171 Onboard Memory

The PXIe-5170/5171 allocates at least 384 bytes of onboard memory for each record in a single multirecord acquisition. Samples are stored in this buffer before transfer to the host computer. Thus the minimum size for a buffer in the onboard memory is four 96-bit samples. Software allows you to specify buffers of less than these minimum buffer sizes but only the specified number of points are transferred from onboard memory into the host computer memory.

The total number of samples that can be stored depends on the acquisition memory size option. The maximum number of records in a single multirecord acquisition is equal to the size of the memory option divided by 192 samples. Refer to the [hardware specifications document](#) for information about the memory options available for each device.

Triggering and Memory Usage

During an acquisition, samples are stored in a circular buffer that is continually rewritten until a trigger is received. These samples are known as the "pre-reference trigger" samples. After the trigger is received, the PXIe-5170/5171 continues to acquire post-reference trigger samples. The relationship of the number of pre-reference trigger samples to post-reference trigger samples is specified by the "Reference position" attribute (by default, 50% of the record length is post-trigger.) The acquired samples are placed into onboard memory. The number of pre-reference trigger or post-reference trigger samples is only limited by the amount of onboard memory.

SMC-Based Device Synchronization



Note This functionality is only available when using NI-SCOPE.

SMC-based oscilloscopes are built on the NI Synchronization and Memory Core (SMC) technology and therefore support TClk synchronization. Refer to the [NI-TClk Synchronization Help](#) for more information.

Related Topics

[Features Supported by SMC-Based Devices and USB Devices](#)

PXIe-5170/5171 Calibration

The PXIe-5170/5171 supports external adjustment of the following parameters:

- DC gain and offset of all DC-coupled paths
- DC offset of all AC-coupled paths
- Sample Clock frequency

Refer to the **NI PXIe-5170R Calibration Procedure** and **NI PXIe-5171R Calibration Procedure** available at ni.com/manuals for more information about calibrating the preceding parameters or information about calibrating additional specifications.

Related Topics

[Summary of Calibration Options](#)

Summary of Calibration Options

The following table summarizes the available calibration options. For more information on a specific option, follow the links in the table.

Calibration	Impact	When to Use	Notes
External calibration	Calibrates DC gain, DC offset, and Sample Clock frequency	Every two years	Calibrates and verifies certain specifications
Self-calibration	Calibrates intermodule synchronization circuitry	90 days, or when temperature changes by $>5^{\circ}\text{C}$	Ensures intermodule synchronization accuracy
No calibration	No impact within 2-year calibration cycle or if temperature stays within $\pm 5^{\circ}\text{C}$	High accuracy not required outside of 5°C	—

Self-Calibration

To maintain intermodule synchronization accuracy, your device provides a highly repeatable self-calibration function.

For the PXIe-5170/5171, self-calibration yields the following benefits:

- Characterizes onboard Sample Clock phase adjustment circuitry.
- Characterizes onboard intermodule synchronization measurement circuitry.
- Takes approximately 15 seconds to complete.

When to Self-Calibrate

For optimum performance, use self-calibration in the following circumstances:

- When the device is placed in a new system and has warmed up for at least 15 minutes. Warm-up begins after the chassis is powered, the device is recognized by the host, and the ADC clock is configured using the instrument design libraries. Running an included sample project or running self-calibration using NI MAX will configure the device and start warm-up.
- Any time the temperature changes more than 5 °C from the previous self-calibration
- 90 days after the previous self-calibration

The result is a product that yields full performance over its operating temperature range and two-year calibration cycle. When the two-year calibration interval expires, an [external calibration](#) is required to ensure performance that is within specification over the next two years.

The PXIe-5170/5171 has a temperature sensor that monitors temperature variations. The device also records the previous self-calibration time and date. For instructions on reading the time and temperature at which self-calibration was last performed, refer to [Calibration Reporting](#).

Running Self-Calibration

To programmatically self-calibrate the NI 5170R/5171R when using instrument design libraries, use the Self Calibrate VI located on the Functions»FPGA Interface»Software-Designed Instruments»Oscilloscopes»NI PXIe-5170R/5171R»Calibration palette.

To programmatically self-calibrate the PXIe-5170/5171 when using NI-SCOPE, use the Self Calibrate VI located on the Functions»Measurement I/O»NI-SCOPE»Calibration palette.

You can also self-calibrate the PXIe-5170/5171 by pressing the Self Calibrate button for the device in MAX.

Input Connections During Self-Calibration

No special care needs to be taken on any front panel terminals during self-calibration.

External Calibration

External calibration involves both verification and adjustment. Verification is the process of testing the device to ensure that the input accuracy is within certain specifications. You can use verification to ensure that the adjustment process was successful. If verification is not successful, perform the adjustment procedure and retry the verification procedure. After a successful verification, update the calibration date using MAX or the Set Verification Date and Time VI.

Adjustment is the process of measuring and compensating for device performance to improve the input accuracy. The device is warranted to meet or exceed its published specifications for the duration of the calibration interval.

National Instruments recommends a calibration interval of two years for the PXIe-5170/5171. Your application may have different calibration interval requirements based on its measurement accuracy demands. You can change the calibration due date with MAX or the Set Calibration Due Date VI.

For the PXIe-5170/5171, external calibration yields the following benefits:

- Corrects for DC gain and offset errors within the oscilloscope by comparison to a precision, high-stability external voltage reference. This is done for all ranges and all filter paths (enabled/disabled).
- Corrects for DC offset errors within the oscilloscope for the AC coupling paths.

- Calibrates internal Sample Clock frequency to ensure accurate sample rate by comparison to a precision, high-stability external reference frequency.

Your device must be externally calibrated at least once every two years in order to maintain adherence to published specifications.

Calibration procedures may be found online at ni.com/manuals.

You can learn more about external calibration at ni.com/calibration.

Calibration Reporting

Reports may be obtained from your device regarding time and temperature at which calibrations were run on your device. To view this information, use the VIs located on the Functions»Software–Designed Instruments»Oscilloscopes»NI PXIe–5170R/5171R»Calibration»Utilities palette.

You can also view this information in MAX and through the NI System Configuration API.

For more information about these VIs and the calibration instrument design libraries, refer to the [Calibration Data](#) and [Calibration API](#) overview pages.

PXIe-5170/5171 Warm-Up

NI recommends warming up the PXIe-5170/5171 for 15 minutes before operation. The unit is fully functional prior to this time, but frequency and amplitude accuracy and other specifications are not guaranteed until the device has fully completed warming up.



Note Warm-up begins after the chassis is powered, the device is recognized by the host, and the device is configured using the instrument design libraries or NI-SCOPE. Running an included sample project or running self-calibration using MAX will configure the device and start warm-up.

PXIe-5170/5171 Power On, Download, and Reset Conditions

The PXIe-5170/5171 applies certain conditions to the device state at power on, device reset, and download. The PXIe-5170/5171 also applies certain conditions to the device state upon FPGA reset.

Power On and Download Conditions

The following conditions apply to the device state after powering on the device, restarting the device, or downloading a new FPGA VI to the device.

- PFI lines are configured as input terminals.
- Signal conditioning peak-to-peak range (V_{pk-pk}) is set to 5 V, coupling is disabled, and filter is set to 100 MHz.
- PXI trigger lines are configured as input terminals.
- Clocking is reset to the component default state.
- External power is enabled and connected directly to the FPGA Window.
- ADCs are reset to the component default state.

Reset Conditions

The following conditions apply to the device state after resetting the device FPGA.

- PFI lines are configured as input terminals.
- Signal conditioning range is maintained.
- PXI trigger lines are configured as input terminals.
- Clock states are maintained.
- External power state is maintained.
- ADC states are maintained.

Integration and System Considerations

This section contains information about integrating digitizers into a PXI/PXIe or PCI based measurement system.

PXI/PXIe Chassis Cooling

Not all PXI or PXI Express chassis provide the same cooling. When selecting a PXI or PXI Express chassis, consideration should be given to providing adequate airflow.

NI PXI and PXI Express devices are high-precision instruments and may be sensitive to interference from other electronic devices. To optimize the accuracy and

performance of the device, you may need to locate the device in a slot away from devices with power supplies and other noisy circuitry. The device may also be sensitive to heat generated by high-power products in neighboring slots. When possible, consider locating the device away from high-power devices to optimize cooling.

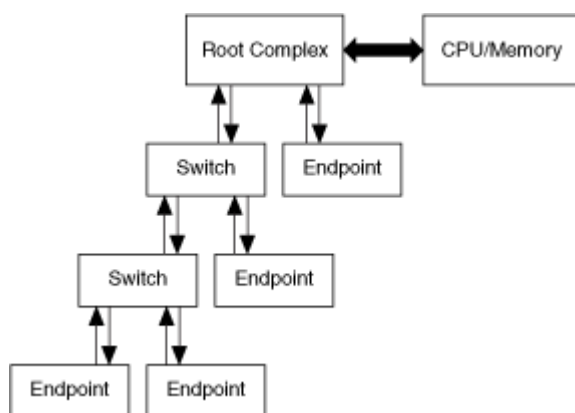
For more information on cooling, refer to the **Maintain Forced-Air Cooling** note that shipped with the device.

PXI Express Topology

PXI Express uses PCI Express signaling and protocols for communication. A PCI Express system topology includes several different types of components:

- **Root Complex**—The root complex is the base of the PCI Express hierarchy that connects the CPU and system memory to PCI Express I/O.
- **Switch**—A switch routes transactions among multiple PCI Express devices, other switches and the root complex. PCI Express communication is always point-to-point. A device communicates to a single switch. The switch then passes on the data to another switch, another PCI Express device, or the root complex.
- **Endpoint**—An endpoint is a device that performs an I/O function, such as a PXI Express module.

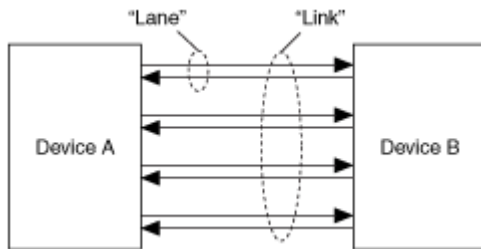
The following diagram shows an example PCI Express system.



Links, Lanes and Training

- **Link**—A link is a communication connection between two PCI Express elements. For example, a switch and an endpoint can form a link. A link consists of one or more lanes.
- **Lane**—A lane is a single transmit/receive pair connection between two PCI Express elements. Two devices can communicate data in both directions on a lane at the same time.

The following figure illustrates a link between two devices consisting of 4 lanes.



The number of lanes in a link is denoted as xn , where n is a number. All PCI Express devices must support $x1$ links. PCI Express devices may also support links of $x2$, $x4$, $x8$, $x12$, $x16$, and $x32$.

The data transfers on a lane can occur at 2.5 Gb/s or 5.0 Gb/s.

During initialization, the two devices of a link must negotiate the lane width and signaling speed that will be used for the link. This process is called link training. All devices must support $x1$ lane widths and 2.5 Gb/s signaling.

Performance

The actual throughput for a given system will be less due to protocol overhead, system topology, data transfers between other devices in the system, and other components in the system.

Transfers between an I/O device and host memory usually travel through several PCIe links. Data transfer performance depends on the performance of each link. For example, if a PXI chassis is controlled with a $x1$ MXI Express link, the throughput to system memory may be limited by the $x1$ link.

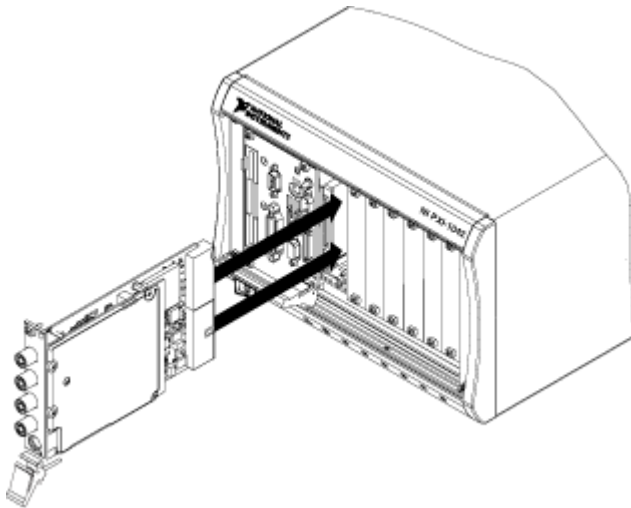
As another example, consider an NI PXIe-1065 chassis. In this chassis, slots 7 and 8 support $x4$ PXI Express devices. Slots 7 and 8 each have a direct $x4$ PCIe path to the PXI controller. Slots 9–14 also support $x4$ PXI Express devices. However, data transfers for all of these slots (as well as PXI transfers for slots 15–18) all travel

through a PCI Express switch on the backplane before transferring to the PXI controller. So all devices in slots 9–18 share the bandwidth available on the link between the backplane switch and the PXI controller.

PXI Modules

PCI eXtensions for Instrumentation (PXI) modular instrumentation delivers a PC-based, standardized, high-performance measurement and automation system. PXI combines the high-speed PCI bus with integrated timing and triggering designed specifically for measurement and automation applications to deliver significant performance improvements over older architectures. For an overview of the PXI Specification, refer to PXI Specification Tutorial.

The following figure shows a typical PXI chassis installation.



Chassis Guidelines

NI PXI digitizers can be installed in the following chassis and slots:

- PXI chassis—PXI modules can be installed in any peripheral slot of a PXI chassis.
- PXI Express chassis—PXI devices can be installed in the following PXI Express chassis slots:
 - PXI-1 slots—Accepts all PXI modules

- PXI hybrid slots—Accepts all PXI modules that are hybrid slot-compatible or PXI Express modules
- PXI Express slots—Accepts PXI Express modules

Using PXI-Compatible Products with Standard CompactPCI Products

The ability to use PXI-compatible products with standard CompactPCI products is an important feature provided by the **PXI Specification**, revision 2.1. If you use a PXI-compatible plug-in device in a standard CompactPCI chassis, you cannot use PXI-specific functions, but you can still use the basic plug-in device functions. For example, the PXI trigger bus on NI signal generators is available in a PXI chassis but not in a CompactPCI chassis. The CompactPCI specification permits vendors to develop sub-buses that co-exist with the basic PCI interface on the CompactPCI bus. Compatible operation is not guaranteed between CompactPCI devices with different sub-buses nor between CompactPCI devices with sub-buses and PXI. The standard implementation for CompactPCI does not include these sub-buses. NI signal generators work in any standard CompactPCI chassis. PXI-specific features, such as PXI_Trig bus and PXI_CLK10 reference are implemented on the J2 connector of the CompactPCI bus.

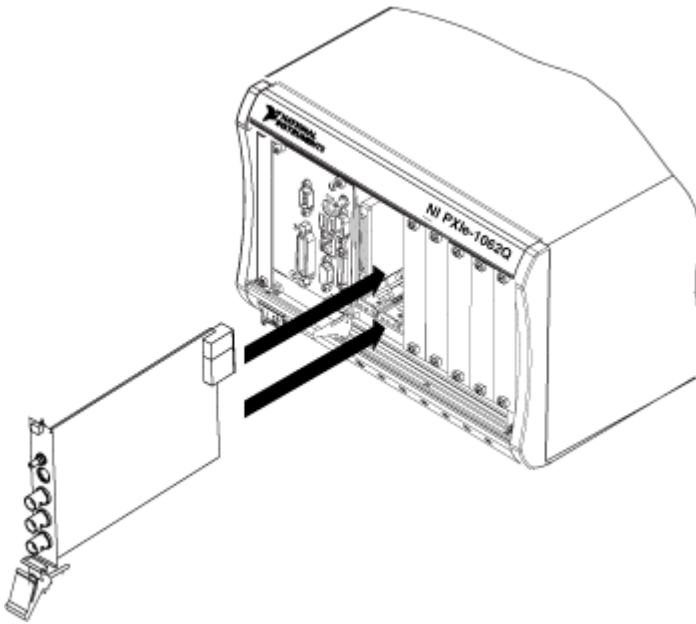
Related Topics

[PXI Star Trigger Line](#)

PXI Express Modules

The PXI Express Specification integrates PCI Express signaling into the PXI standard, which increases backplane bandwidth and enhances PXI timing and synchronization features by incorporating a 100 MHz differential reference clock and differential triggers. The PXI Express Specification adds these features to PXI while maintaining backward compatibility. For an overview of the PXI Express specification, refer to PXI Express Specification Tutorial.

The following figure shows a typical PXI Express chassis installation.



Chassis Guidelines

NI PXI Express modules can be installed in the following PXI Express chassis slots:

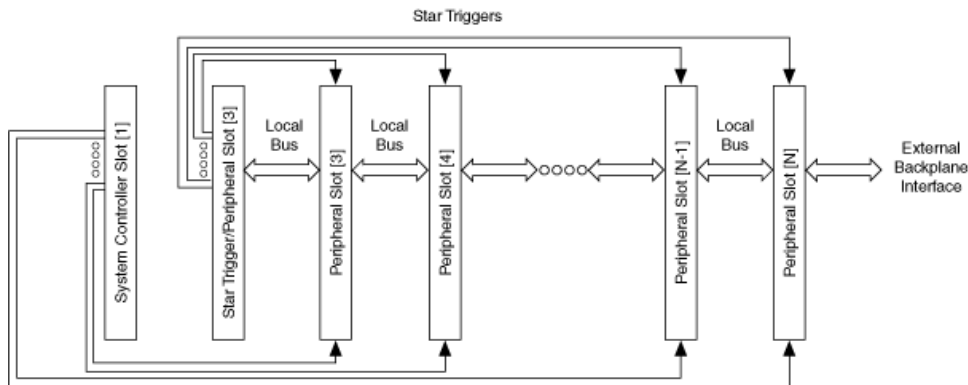
- PXI hybrid slots—Accepts all PXI modules that are hybrid slot-compatible or PXI Express modules
- PXI Express slots—Accepts PXI Express modules



Note Refer to the documentation for your PXI Express chassis for more information about installing and configuring PXI Express modules.

PXI Star Trigger Line

The PXI star trigger is a feature implemented on National Instruments PXI chassis. PXI chassis have a PXI trigger bus that is linked to all slots in the chassis. In addition, PXI chassis have a star trigger that is linked to Slot 2. The star trigger is a high-performance trigger signal that you can use to synchronize all the modules in a chassis. You can also do this using the normal PXI trigger bus, but the star trigger offers increased performance, specifically a propagation delay of no more than 5 ns and an intermodule delay of no more than 1 ns.



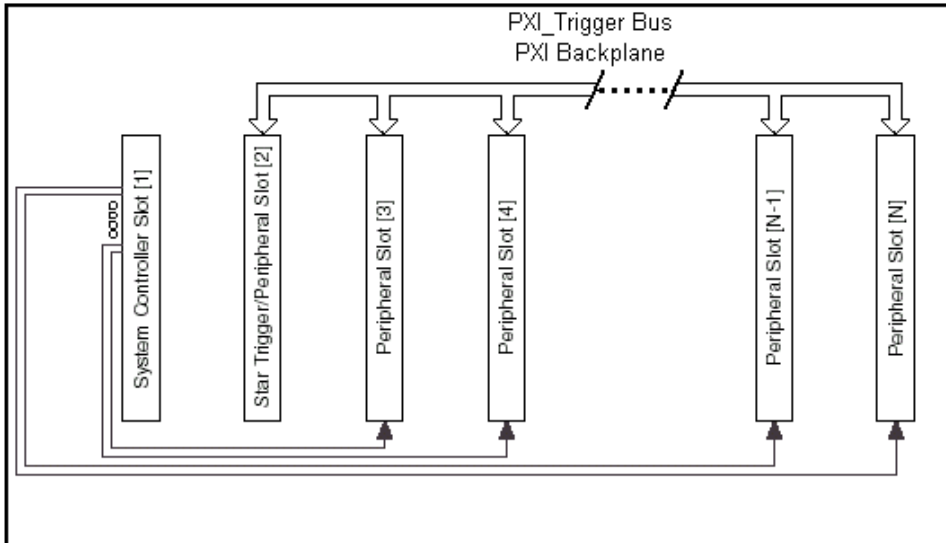
Slot 2 of the PXI chassis is optimized for a star trigger controller. If the star trigger is not required, you can use Slot 2 as a standard peripheral slot. The trigger signal generated by a controller can then be accessed by the other modules through the backplane. The star trigger controller in Slot 2 of the PXI chassis can route triggers among peripheral slots with very low skew (within 1 nanosecond).

To use a star trigger, the digitizer must be in a PXI form factor.

All NI digitizers can receive triggers from the PXI star trigger line, which means that if another instrument is driving the PXI star trigger from Slot 2 of the PXI chassis, the NI digitizer can receive this trigger signal from another slot. NI digitizers cannot drive the PXI star trigger line from Slot 2. Therefore, if you want to generate a trigger on the star trigger line with a master in Slot 2, this master cannot be an NI digitizer.

PXI Trigger Lines

Eight PXI based trigger lines are highly flexible and can be used in a variety of ways. For example, triggers can be used to synchronize the operation of several different PXI peripheral modules. In other applications, one module can control carefully timed sequences of operations performed on other modules in the system. Triggers may be passed from one module to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled. The number of triggers that a particular application requires varies with the complexity and number of events involved.



The PXI Specification is implemented with the RTSI bus through the PXI trigger lines. PXI Specification requires 8 lines, PXI_TRIGGER[0:7], on the P2/J2 connector of the PXI chassis for the trigger lines. The RTSI features of the NI Source hardware is implemented on this sub-bus. The RTSI trigger [0..6] is implemented on PXI_TRIGGER[0:6], and the RTSI clock is routed on PXI_TRIGGER(7).

System Reference Clock (PXI_CLK10)


The PXI chassis supplies the PXI 10 MHz system clock signal (PXI_CLK10) independently to each peripheral slot. An independent buffer drives the clock signal to each peripheral slot. The buffer has a source impedance matched to the backplane and a skew ranging from less than 1 ns to more than 250 ps between slots. You can use this common reference clock signal to synchronize multiple modules in a measurement or control system. You can drive PXI_CLK10 from an external source through the PXI_CLK10_IN pin on the P2 connector of the star trigger slot. Sourcing an external clock on this pin automatically disables the backplane's 10 MHz source. You can synchronize multiple chassis that have connectors on the back panel for 10 MHz Reference In and 10 MHz Reference Out. Refer to your PXI chassis documentation for more information.

PFI Lines

PFI lines are multipurpose programmable function input/outputs. These lines serve as connections to virtually all internal timing signals. Many NI high-speed digitizers

have two or more digital lines that can import a trigger, import or export a reference clock, or output various signals.

The function of each PFI line is independent. However, only one trigger source can be imported during acquisition.

 **Notice** If you enable a PFI line for output, do not connect any external signal source to it; doing so can damage the device, the computer, and the connected equipment.

MXI Optimization Application

MXI-3

If you are using the MXI-3 interface to control the PXI chassis, the MXI-3 Optimization Application must be run prior to using the NI high-speed digitizer. By default, this application runs automatically when Windows starts. If you have an initialization, timeout, or performance issue with your module, or if you are not certain that the application ran, select **Start»Programs»National Instruments MXI-3»MXI-3 Optimization** to run the application. If you continue to have initialization or performance issues, refer to the MXI-3 documentation at **Start»Programs»National Instruments MXI-3**, or visit NI Technical Support at ni.com/support.

MXI-4 and MXI-Express Optimization

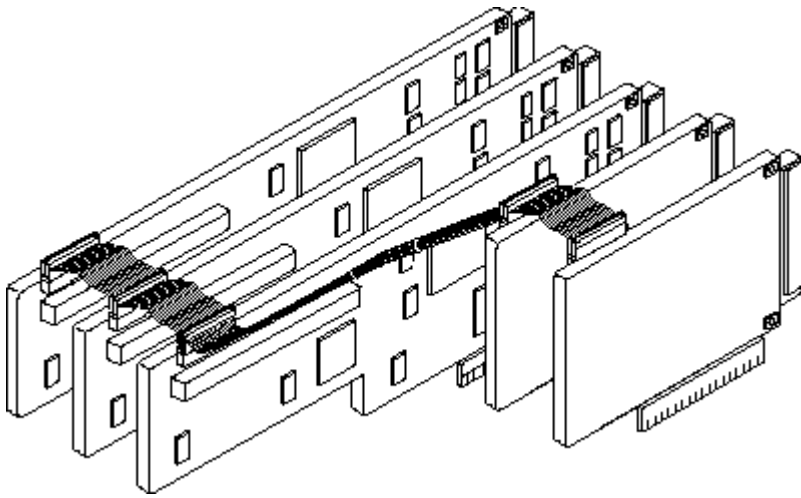
Optimization for MXI-4 and MXI Express are performed automatically by the hardware.

RTSI Bus

RTSI stands for Real-Time System Integration. It is a bus found on many National Instruments devices that, when cabled together with a RTSI cable, is used to share and exchange timing and control signals between multiple boards. It is usually used for synchronization purposes.

The RTSI bus cables are short, 34-conductor ribbon cables equipped with two to five connectors to link together a group of boards. The cable options include connections for two, three, four, and five boards, plus an extended cable length to

connect up to five long and short boards. The following figure shows an example of an extended five-board cable setup.



PFI Lines

PFI lines are multipurpose programmable function input/outputs. These lines serve as connections to virtually all internal timing signals. Many NI high-speed digitizers have two or more digital lines that can import a trigger, import or export a reference clock, or output various signals.

The function of each PFI line is independent. However, only one trigger source can be imported during acquisition.



Notice If you enable a PFI line for output, do not connect any external signal source to it; doing so can damage the device, the computer, and the connected equipment.

NI-SCOPE Soft Front Panel and InstrumentStudio

October 2018

When you install NI-SCOPE on a 64-bit system, you can monitor, control, and record measurements from supported devices using InstrumentStudio. InstrumentStudio is a software-based front panel application that allows you to perform interactive

measurements on several different device types, including oscilloscopes, in a single program.



Note InstrumentStudio support was first available in NI-SCOPE 18.1. InstrumentStudio is supported only on 64-bit systems. If you are using a 32-bit system, use the [NI-SCOPE Soft Front Panel](#).

InstrumentStudio is automatically installed when you install the NI-SCOPE driver. You can access InstrumentStudio in one of the following ways:

- From the Windows start menu, select National Instruments»NI-SCOPE Soft Front Panel. This launches InstrumentStudio and runs a soft front panel populated with NI-SCOPE devices.
- From the Windows start menu, select National Instruments»InstrumentStudio [**year**]. This launches InstrumentStudio and runs a soft front panel populated with devices detected on your system.
- From Measurement & Automation Explorer (MAX), select a device and then click Test Panels.... This launches InstrumentStudio and runs a soft front panel for the device you selected.

For more information on using InstrumentStudio, refer to the [InstrumentStudio Manual](#).

© 2001–2018 National Instruments Corporation. All rights reserved.

Related Documentation

Refer to the National Instruments Product Manuals Library at ni.com/manuals for updated documentation resources.

The following documents contain information that you may find helpful as you use this help file:

- **NI High-Speed Digitizers Help**

NI-SCOPE 32-Bit Soft Front Panel Help

This help file contains information about using the 32-bit NI-SCOPE Soft Front Panel (SFP) to interactively control your NI high-speed digitizer. For information about developing applications, refer to [NI High-Speed Digitizers Help»Programming](#).



Note The information in this help file is relevant only to the NI-SCOPE SFP that launches in NI-SCOPE 18.0 and earlier or that, beginning with NI-SCOPE 18.1, launches on 32-bit systems. For information on InstrumentStudio, a software-based front panel application for 64-bit systems that was first available in NI-SCOPE 18.1, refer to [NI-SCOPE Soft Front Panel and InstrumentStudio](#).

Similar to stand-alone oscilloscopes, the SFP acquires, controls, analyzes, and presents data. Unlike bench-top instruments, however, you can use the SFP to view and control waveforms directly from your computer. You can also run multiple sessions of the SFP simultaneously.

The SFP provides the following functionality:

- Easily observe and control signal
- Debug a device
- Load/store waveforms
- Load/store device-specific configurations
- Make scalar measurements
- Use NI-SCOPE utilities such as Auto-Setup and Self-Calibration



Note Most features of the SFP are documented in the SFP context help. You can view the context help for a SFP element by hovering the cursor over the element you want more information about. If the context help window does not appear, press <Ctrl-H> to activate the context help, or select Help»Show Context Help.

Accessing the NI-SCOPE Soft Front Panel

You can access the SFP two ways:

- Launch the SFP by selecting Start»All Programs»National Instruments»NI-SCOPE»SCOPE Soft Front Panel or from NI Launcher in Windows 8.

- Navigate to `<IVIROOTDIR32>\Drivers\niScope\NI-SCOPE_Soft Front Panel.exe`. `<IVIROOTDIR32>` is an alias to a specific National Instruments file folder location. Refer to the **NI-SCOPE Readme** for more information about installed file locations.

Selecting or Simulating a Device

You must either connect to or simulate a device to use the SFP. If you have already registered an active device or simulated a device in Measurement & Automation Explorer (MAX), you can choose from a list of available devices.

If you do not have any active digitizers registered in MAX, the SFP will prompt you to simulate a device. Choose the Model Number and Bus Type of the device you want to simulate, then select the amount of Noise (%) to simulate. The SFP will simulate data acquired from the type of digitizer you have chosen.



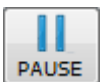
Note Monitoring or controlling a simulated device may result in unexpected data.

Important SFP Elements

The SFP is designed to look similar to the front panel of a desktop instrument. However, some functionality of the SFP is fundamentally different from that of a box oscilloscope.

- The SFP does not generate data.
- The SFP configures a connected or simulated device and acquires, analyzes, and presents data from that device.
- The SFP can control a device in use by another application during a Debug Driver session.

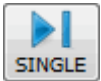
The following buttons control the connection and data acquisition of your device.



The Pause button pauses the acquisition. The device is still connected to the SFP, but other programs are not able to acquire data from the device. The Pause button becomes the Run button while the acquisition is paused.



The Run button resumes a paused acquisition. It also prompts you to select a new device if you have released a previous session with a device.



The Single button runs one acquisition.



The Off button stops the acquisition. When stopped, all drivers are unloaded. Select the Run button to resume an acquisition.



The Auto button automatically configures the device. The device recognizes the input signal and automatically configures many instrument settings.



The Prop button opens a dialog box and displays all properties available in a session.

Control Tabs



The control tabs on the SFP are similar to the soft buttons on a box oscilloscope. The control tabs are, from left to right: Channel Settings (CHAN), Trigger Settings (TRIG), Horizontal Settings (HORIZ), Measurement Settings (MEAS), Cursor Settings (CURS), and Display Settings (DISP). The Trigger Settings tab is active by default.

Related Topics

[Sharing Control of a Device](#)

Monitoring a Device

Debug Driver Session in the Soft Front Panel

In a Debug Driver session, you can monitor or control your device in the Soft Front Panel (SFP) while the device is being used by another application, such as LabVIEW or TestStand. Changing device settings in the SFP can help you troubleshoot problems with your device under test or application.



Note The SFP automatically opens in a Debug Driver session for a device that is In Use.

Example

You want to change the device settings based on unexpected results from your application. You immediately pause the application where you suspect an error, start the SFP and choose the device for which you want to change settings. You can change device settings, then resume your application with the new settings.

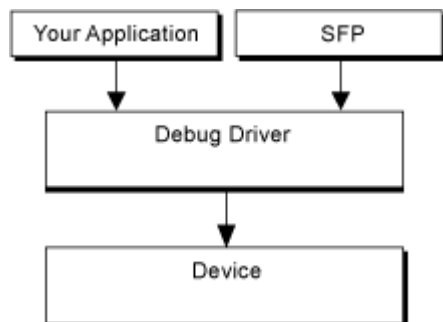
Related Topics

[Sharing Control of a Device](#)

[Monitoring a Device](#)

Debug Driver Session Model

The following figure shows a diagram of a typical Debug Driver session.



Debugging a Device in the SFP While in Use by Another Application

You can monitor or control your device in the SFP when the device is in use by other applications, such as LabVIEW or TestStand. Select **Debug Driver Session (Monitor)** from the pull-down menu to view data and device settings. Most SFP controls are disabled because the SFP is read-only when **Debug Driver Session (Monitor)** is selected. You cannot make changes in the SFP that will affect your device or application in this mode.

Select **Debug Driver Session (Control)** from the pull-down menu if you want to change device settings. The SFP is read-write when **Debug Driver Session (Control)** is selected. Controls are enabled and you have full access to your device. To relinquish control, select **Debug Driver Session (Monitor)** or resume execution of your application.

Related Topics

[Sharing Control of a Device](#)

[Monitoring a Device](#)

Sharing Control of a Device

You can debug your device by sharing a device between the SFP and another application.

Controlling a Device in Use by LabVIEW

Complete the following steps to share control between the SFP and LabVIEW:

1. Transfer control of the device from LabVIEW to the SFP.
 1. Create a probe on a niScope session wire at a point where you suspect device setting errors. For more information about probes, refer to LabVIEW documentation.
 2. Select **Debug in Soft Front Panel** from the **Probe Watch Window**.

3. Leave the Probe Watch Window open, and run your code. Code execution pauses at the probe, and the SFP opens with Debug Driver Session (Control) selected.
2. Change the device settings in the SFP manually.
3. Select Debug Driver Session (Monitor) from the pull-down menu, or close the SFP. Control of the device is given back to LabVIEW.
4. Apply your changed device settings to your LabVIEW application, or revert your changes in the Scope Session Changes dialog box before closing the SFP.



Tip Select Show Changes in the Debug Session Changes dialog box to display a log of device setting changes made during the Debug Driver session. Select Generate VI in the dialog box to create a VI of the device setting changes you made during the session.

5. (Optional) Continue using your device in your LabVIEW application, with your new device settings.

Controlling a Device in Use by CVI, TestStand, or Another Application

Complete the following steps to share control between the SFP and CVI, TestStand, or another application:

1. Pause the external application that is using your device.
2. Launch the SFP.
3. Select Debug Driver Session (Control) from the pull-down menu in the SFP.
4. Change the device settings in the SFP manually.
5. Select Debug Driver Session (Monitor) from the pull-down menu, or close the SFP. Control of the device is given back to your application.
6. Apply your changed device settings to your application, or revert your changes in the Scope Session Changes dialog box before closing the SFP.



Tip Select Show Changes in the Debug Session Changes dialog box to display a log of device setting changes made during the Debug Driver session. Select Export as Text to document the changes.

7. (Optional) Continue using your device in the other application, with your new device settings.

Monitoring a Device

You can monitor a device with LabVIEW, CVI, TestStand, or your application in the SFP to troubleshoot problems with your device under test or application. View data and device settings by sharing your device between the SFP and your application.

Complete the following steps to monitor a device with the SFP:

1. Launch the SFP.



Tip You can launch the SFP from the Windows Start menu, the Tools menu in LabVIEW, or from your device listed under Devices and Interfaces in the MAX configuration tree.

2. Select an In Use device that you want to monitor with the SFP. Debug Driver Session (Monitor) is selected from the pull-down menu.
3. When you finish monitoring the device, close the SFP.

Troubleshooting a Debug Session in the SFP

Complete the following to correct common issues you might encounter while debugging your device.

Why did I lose control of the device in the SFP?

Your application made a call to the NI-SCOPE driver. Pause your external application and select Debug Driver Session (Control) from the pull-down menu, or ensure that the application does not fetch data or access the device.

How do I debug my C application? I get an error when I debug my C application.

From the SFP, select Utility»Configure Debug Session»Using Breakpoints in C/C++/.NET? to debug your C application.

Why is my performance slow?

Supporting communication with your device in C/C++/.NET environments at breakpoints requires additional overhead that affects performance.

Selecting **Using Breakpoints in C/C++/.NET?** may cause a slight performance impact. When you complete debugging your C application, deselect **Using Breakpoints in C/C++/.NET?** in the SFP. Restart your application to implement the changes.

Why can't I select a device that is "In Use"?

The SFP is currently monitoring or controlling a shared device session, or the device does not support debugging. Verify that no external applications are currently accessing the device.

How do I enable/disable my device for a Debug Session?

Complete the following steps in the SFP to ensure the device is enabled for Debug Session mode (the default value is enabled).

1. Select **Utility»Configure Debug Session** and click the device in use.
2. Select **Enabled** in the **Debug Session Configure** dialog box.
3. (Optional) Select **Using Breakpoints in C/C++/.NET?** if you are setting breakpoints in a C/C++/.NET application.

Saving Waveform Data

Select **File»Save Data** to open the File Save dialog.

Waveform data can be saved in three file formats: [hierarchical waveform storage \(*.hws\)](#) data file, [LabVIEW Measurement \(*.lvmm\)](#) data file, and [flat binary \(*.bin\)](#). Waveforms saved with the NI-SCOPE Soft Front Panel can be generated using an NI arbitrary waveform generator and the NI-FGEN Soft Front Panel.

Shortcut:

<Ctrl-S>

Hierarchical Waveform Storage Data File (.hws)

Hierarchical Waveform Storage Data File (.hws) format is based on the HDF5 file format, which is a widely used and efficient file format for storing scientific and

engineering data such as images, vector arrays, and structured and unstructured grids. For more information about HDF5, refer to the HDF Group web site at <http://www.hdfgroup.org/HDF5/>.

HWS is the file type of choice due to its flexibility. It is capable of saving device configuration settings as well as multiple channels of waveform data, measurements, and associated header information. HWS also offers compression options, making this format the best choice for very large or multiple data records.

LabVIEW Measurement Data File (.lvm)

The LabVIEW Measurement (.lvm) format is a text-based file format for one-dimensional data that you can use with the Read LabVIEW Measurement File and Write LabVIEW Measurement File Express VIs.

The .lvm format is easy to parse and easy to read when imported into a spreadsheet program such as Microsoft Excel, or a text editor such as Notepad. The .lvm format supports multiple data sets, grouping of data sets, and the addition of data sets to existing files.

This file format is not designed for high-performance or for very large data sets, as is the case with all text-based formats.



Note For very large data sets, use the [HWS](#) or [binary](#) file formats.

For more information on the .lvm file format, refer to the Specification for the LabVIEW Measurement File.

Binary Waveform Data Format (.bin)

The NI-SCOPE Soft Front Panel can save data in a binary file that conforms to the following specifications:

- File extension—.bin
- File header—none
- Data—Data are stored as a continuous single stream. The creator of the file must remember the data type the data were stored in. Data can be imported in 8-, 16-, or 32-bit binary, and can be little or big endian format.

Binary files save only one channel of data and offer advanced options for byte format. Associated header information is saved in a separate text file with the same name.

Saving Configuration Settings

Configuration settings are saved in three ways: automatically upon stopping or closing the SFP, using the File»Save Configuration option, and when saving data in [HWS](#) file format.

The SFP always saves your device configuration settings upon release of a session or closing of the SFP. If the SFP detects the same device using the same resource descriptor (set in MAX) upon reopening, it loads the previous configuration settings. If you are using a different type of device, the SFP configures your acquisition automatically using the Autosetup function.

To save your configuration settings, select File»Save Configuration. Enter the name of your configuration file or select an existing configuration file and click Save.



Note Several configurations can be saved in one configuration file. Internal to the saved file, each configuration is distinguished by an indexed unique group name. As such, you can have two configurations with the same label in the same file. The same label will not overwrite a previous configuration.

Enter any additional information you would like to add in the fields provided. Click OK to add your current configuration to the selected file.

Configuration settings can also be [saved along with waveform data](#).

Backing Store

Backing store is a feature that stores the current waveform on the NI-SCOPE Soft Front Panel in RAM. During normal operation, the NI-SCOPE SFP gets the data from the digitizer, decimates and scales it for display, displays it, then discards it. This process saves time and memory, especially when working with large data sets. With large data sets, the data is never all in memory at one time, because it is read from the digitizer in chunks and decimated in chunks. The NI-SCOPE SFP scales this decimated data. The disadvantage of this process is that you must wait for an acquisition to finish before you get data that you can save. If you have a signal on

the NI-SCOPE SFP that you want to save, you may not be able to do so unless the acquisition is currently paused, depending on the length of the waveform.

To save a waveform, the NI-SCOPE SFP copies chunks of data from the digitizer directly to disk, which prevents a memory hit. If your acquisition is in progress, the data that is currently displayed is lost, and you must wait until the acquisition is complete before saving data. Using single-shot mode is usually recommended if you want to save the displayed data. When the digitizer pauses, the data is preserved because another acquisition is not initiated. If backing store is enabled, the current waveform is stored in memory and you can save at any time.

Backing store is enabled by default. If you are using an older computer, however, you may experience reduced performance and higher memory requirements. Disabling backing store may solve these problems.

Keyboard Shortcuts



Note Shortcuts for controls grouped under a Control Tab are only usable while their Control Tab is active.

Shortcut	Description
<G>, <R>, <Shift-G>, <Shift-R>	Run
<P>, <Shift-P>	Pause
<G>, <R>	Resume
<ESC>, <Q>, <Shift-Q>	Stop
<Shift-X>, <X>, <Ctrl-Q>, <Alt-F4>	Exit
<A>, <Shift-A>	Auto setup
<1>	Trigger a single-shot acquisition
<)> (usually Shift-0)	Set left channel selector to channel 0
<!> (usually Shift-1)	Set left channel selector to channel 1
<#> (usually Shift-3)	Set left channel selector to channel 2 (if available)
<\$> (usually Shift-4)	Set left channel selector to channel 3 (if available)
<Ctrl-)> (usually Ctrl-Shift-0)	Set right channel selector to channel 0
<Ctrl-!> (usually Ctrl-Shift-1)	Set right channel selector to channel 1

<Ctrl-#> (usually Ctrl-Shift-3)	Set right channel selector to channel 2 (if available)
<Ctrl-\$> (usually Ctrl-Shift-4)	Set right channel selector to channel 3 (if available)
<Ctrl-D>	Enable/Disable the left channel
<Ctrl-Shift-D>	Enable/Disable the right channel
<Ctrl-up arrow>	Decrease volts per division on the left channel
<Ctrl-down arrow>	Increase volts per division on the left channel
<Ctrl-Shift-up arrow>	Decrease volts per division on the right channel
<Ctrl-Shift-down arrow>	Increase volts per division on the right channel
<Ctrl-A>	Increment coupling on left channel
<Shift-Ctrl-A>	Increment coupling on right channel
<Up arrow>	Increase left channel software offset
<Down arrow>	Decrease left channel software offset
<Shift-up arrow>	Increase right channel software offset
<Shift-down arrow>	Decrease right channel software offset
<Shift-up arrow>	Increase vertical position slider
<Shift-down arrow>	Decrease vertical position slider
<Ctrl-M>	Launch measurement window
<Ctrl-D>	Launch device configuration window
<Ctrl-left arrow>	Decrease horizontal range
<Ctrl-right arrow>	Increase horizontal range
<F2>	Change reference position to 10%
<F3>	Change reference position to 50%
<F4>	Change reference position to 90%
<Shift-right arrow>	Increase reference (trigger) position slider
<Shift-left arrow>	Decrease reference (trigger) position slider
<Y>	Increment acquisition type
<Shift-Y>	Decrement acquisition type
<Ctrl-E>	Display error info
<Ctrl-O>	Turn on/off probe compensation
<Ctrl-T>	Turn on/off trigger cursor
<Ctrl-Shift-S>	Run self-calibration

<Ctrl-P>	Print window
<Ctrl-X>	Cut
<Ctrl-V>	Paste
<Ctrl-H>	Show context help
<Ctrl-S>	Save waveform data
<E>	Increment EXT coupling
<Shift-E>	Decrement EXT coupling
<C>	Toggle Cursors
<F1>, <Ctrl-?>	Contents and index
Trigger Settings (TRIG) Control Tab	
<S>	Increment trigger source
<Shift-S>	Decrement trigger source
<T>	Increment trigger type
<Shift-T>	Decrement trigger type
<U>	Increment trigger coupling
<Shift-U>	Decrement trigger coupling
<L>	Increment trigger slope
<Shift-L>	Decrement trigger slope
<M>	Increment trigger mode
<Shift-M>	Decrement trigger mode
<W>	Increment trigger window mode
<Shift-W>	Decrement trigger window mode
<Ctrl-W>	Increment trigger delay
<Ctrl-Shift-W>	Decrement trigger delay
<PgUp>	Increment trigger level/low level
<PgDwn>	Decrement trigger level/low level
<Shift-PgUp>	Increment trigger hysteresis/high level
<Shift-PgDwn>	Decrement trigger hysteresis/high level
<V>	Increment TV event
<Shift-V>	Decrement TV event
<O>	Increment TV polarity
<Shift-O>	Decrement TV polarity

<N>	Increment TV line number
<Shift-N>	Decrement TV line number
<F>	Increment TV signal format
<Shift-F>	Decrement TV signal format
<Ctrl-R>	Toggle TV DC Restore
Horizontal Settings (HORIZ) Control Tab	
<Z>	Enable zoom
Display Settings (DISP) Control Tab	
<Ctrl-M>	Toggle show/hide measurement/cursor data
<Ctrl-C>	Toggle show/hide configuration data
<H>	Hide all data

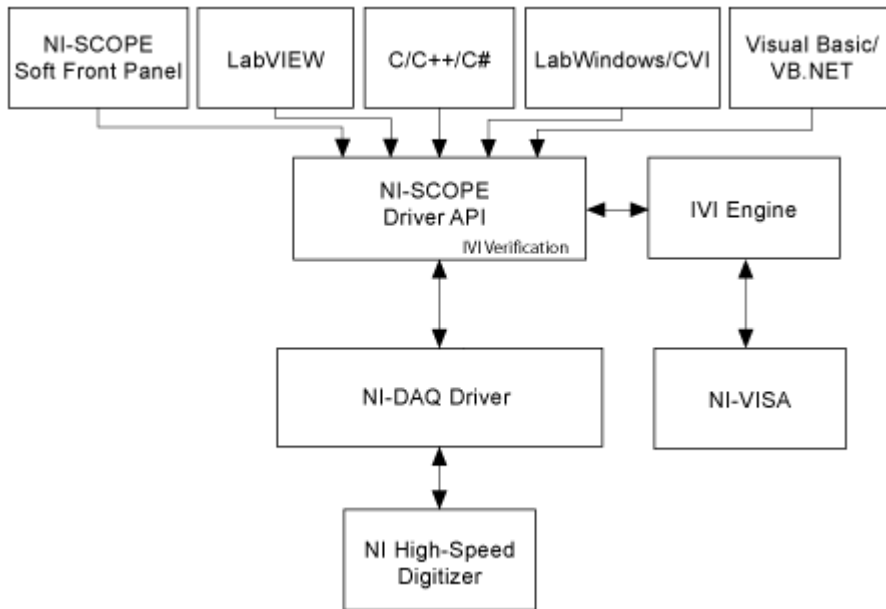
Programming

This section contains information on programming NI high-speed digitizers. You can acquire data with an NI high-speed digitizer either programmatically or interactively.

Programmatically Acquiring Data using NI-SCOPE.

If you want to integrate a digitizer into your test and measurement application, you can program the device using National Instruments LabVIEW or LabWindows/CVI, C/C++, or Microsoft Visual Basic.

The following figure illustrates the relationship between your programming environment, NI-DAQmx, and the digitizer.



Interactively Acquiring Data with the NI-SCOPE Soft Front Panel

If you are using the digitizer as a general-purpose digitizer, you can use the NI-SCOPE Soft Front Panel (SFP) to make measurements interactively without writing code. The NI-SCOPE SFP is a software application for NI high-speed digitizers. The NI-SCOPE SFP is a full-featured, easy-to-use interface with 40 waveform measurements that you can use immediately without learning a programming language.

To launch the NI-SCOPE SFP, go to Start»All Programs»National Instruments» NI-SCOPE»NI-SCOPE Soft Front Panel.

For more information, refer to the [NI-SCOPE Soft Front Panel and InstrumentStudio](#).

Getting Started with NI-SCOPE

NI-SCOPE is both the application programming interface (API) and the [instrument driver](#) that controls your NI high-speed digitizer. NI-SCOPE is included with all NI high-speed digitizers.

Features

NI-SCOPE has the following features:

- Complies with the IVI-Scope driver class specification
- Handles continuous acquisition and multiple device synchronization
- Seamlessly integrates with National Instruments LabVIEW and LabWindows/CVI, as well as conventional programming languages such as C/C++ and Visual Basic
- Includes more than 50 built-in measurement and analysis functions
- Processes waveform measurements in software rather than in hardware

The NI-SCOPE driver provides flexibility and programmability in a standard full-featured instrument driver format.

Examples

To help you get started, NI-SCOPE comes with examples you can use or modify. You can find shortcuts to the examples for each ADE at **Start»All Programs»National Instruments»NI-SCOPE»Examples**.

NI-SCOPE Programming Flow

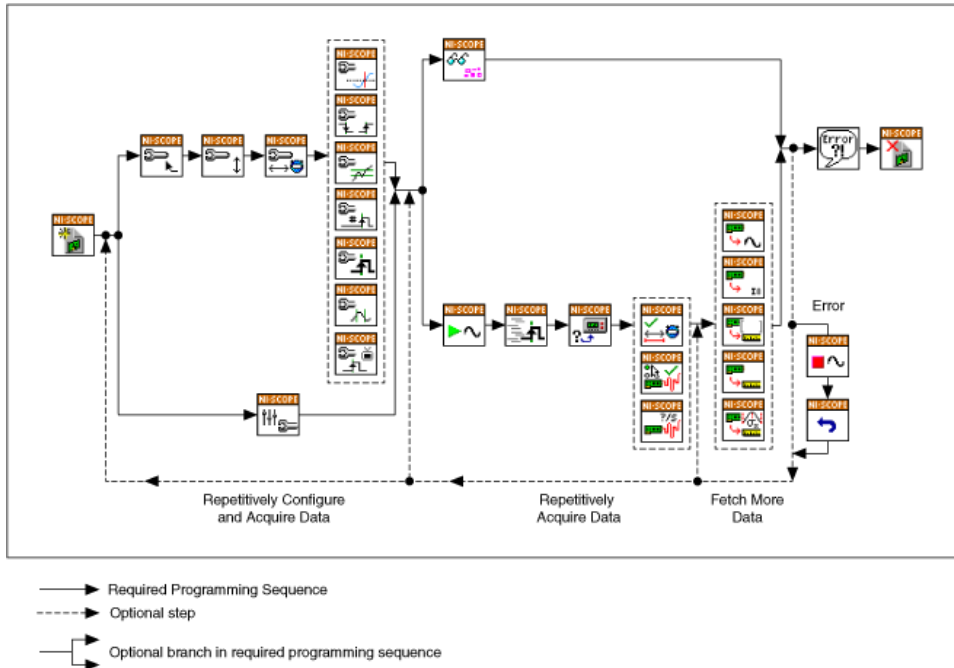
NI-SCOPE provides the same functionality in two formats:

- In LabVIEW as VIs
- In traditional programming languages as functions



Note If you are new to NI-SCOPE, read [Getting Started with NI-SCOPE \(Tutorial\)](#) before you read this section.

The following diagram illustrates the basic programming flow for using NI-SCOPE functions in your own digitizer applications. Click the VIs in the diagram for more detailed programming information.



For any application you write, you must open a session to establish communication with the digitizer(s) by using [niScope Initialize](#) or [niScope Initialize With Options](#). When your program finishes, you must close the session with [niScope Close](#).

Initialize functions set the driver and digitizers to a known state. This function may take a significant amount of time compared to all other NI-SCOPE functions, so you should not include it in a loop when repeatedly acquiring data. Ideally, your program should call an Initialize function one time. If the `reset device` parameter is set to `TRUE`, the digitizer resets to the default state, which may include resetting relays and resetting time stamp counters.



Notice Resetting digitizers may cause wear on the relays, so you should reset only when necessary.

[niScope Close](#) is essential for freeing resources, including deallocating memory, destroying threads, and freeing operating system resources. You should close every session that you initialize, even if an error occurs during the program. While debugging your application, it is common to abort execution before you reach `Close`. Though aborting the execution should not cause problems, NI does not recommend doing so.

Examples

All NI-SCOPE examples include [niScope Initialize](#) or [niScope Initialize With Options](#), [niScope Close](#), and [Error Message](#) (except for LabVIEW). The [Getting Started](#) example is a good choice to start with if you are inexperienced with NI-SCOPE because it is the simplest example.

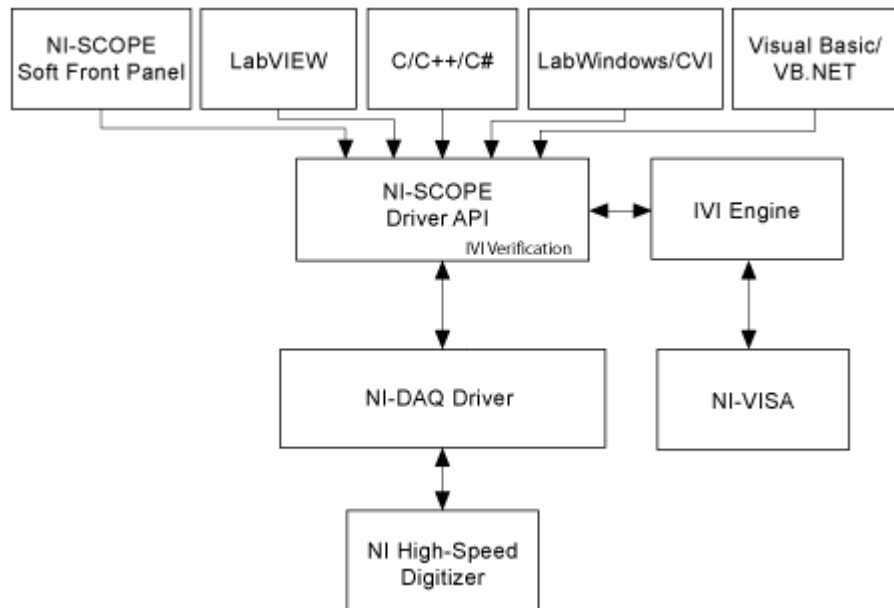
Acquiring Data with NI High-Speed Digitizers

You can acquire data with an NI high-speed digitizer either programmatically or interactively.

Programmatically Acquiring Data

If you want to integrate a digitizer into your test and measurement application, you can program the device using National Instruments LabVIEW or LabWindows/CVI, Microsoft Visual C++/C#, Microsoft Visual Basic, and Microsoft Visual Basic .NET.

The following figure illustrates the relationship between your programming environment, NI-DAQmx, and the digitizer.



Interactively Acquiring Data

If you are using the digitizer as a general-purpose digitizer, you can use the NI-SCOPE Soft Front Panel (SFP) to make measurements interactively without writing code. To launch the NI-SCOPE SFP, select `Start»All Programs»National Instruments»NI-SCOPE»NI-SCOPE Soft Front Panel`. For more information, refer to the [NI-SCOPE Soft Front Panel Help](#).

Acquisition Functions

Reading versus Fetching

You can acquire data by calling either a Read function or a Fetch function. The Read functions are the easiest way to acquire data from your digitizer. They initiate an acquisition, wait for it to complete, and retrieve the data.

Fetch functions assume that the acquisition has already been initiated. Both Read and Fetch functions accept a timeout parameter and a number of points to retrieve. They will efficiently sleep while waiting for the number of requested points to be acquired.

Using a Fetch function in conjunction with the acquisition status function allows you to perform other operations while waiting for the acquisition to finish. When acquisition status reports the operation is complete, you can call the Fetch function with any timeout value to retrieve the data.

You must use Fetch functions to acquire binary data because Read functions allow retrieval of scaled data only. Fetching binary data is faster and requires less user memory than fetching scaled data.

If you want to use a software trigger to start the acquisition, you must use a Fetch function because Read functions block until the operation is complete.

Refer to [Fetching Data](#) for more information on using Fetch functions.

Reading and Fetching Examples

The `Getting Started` example uses a Read function. Most examples use Fetch functions, but the `Binary Acq` example provides the least complicated source

code. You can find shortcuts to these examples at Start»All Programs»National Instruments»NI-SCOPE»Examples.

Configuration Functions

Use the Configuration functions to set up your acquisition.

Using these functions, you can set triggers, input impedance, DC offset, vertical range, sampling rate, and much more. Additionally, NI-SCOPE can automatically configure your device settings with the `Auto Setup` function. Once configured, you can also export and import device settings between matching models.

Configuring Different Parameters

You can use the Configuration functions to configure the following settings and parameters:

- [Acquisition type](#)
- [Vertical settings](#)
- [Horizontal settings](#)
- [Configuring triggers](#)

Configuration Example

The `Configured Acquisition` example demonstrates the Configuration functions. This example supports most of the functionality of NI-SCOPE, so you can experiment with the digitizer.

Configuring the Acquisition Type

NI digitizers support different acquisition types. You set the acquisition type with the `niScope Configure Acquisition VI` or the `niScope_ConfigureAcquisition` function. The different acquisition types change the processing of the data that is returned. In conventional (or normal) mode, the digitizer acts like a traditional instrument, measuring voltage waveforms. In [PXI/PCI-5922 flexible resolution mode](#), the digitizer uses an advanced averaging algorithm to increase the resolution of data at lower sample rates.



Note Not all digitizers support flexible resolution mode. Refer to [Features Supported by Device](#) for more information. By default, the acquisition type parameter is set to normal, and you can omit `Configure Acquisition` if you do not want to change the acquisition type.

Configuring the Vertical Settings

Both `Configure Vertical` and `Configure Chan Characteristics` affect settings that may be different for each channel. Therefore, if you have a two-channel digitizer, you can call each of these functions twice, changing the channel name parameter between 0 and 1. If you want to set these parameters to the same value for multiple channels, use a channel list such as 0,1.

With the `Configure Vertical` function, you can adjust the vertical range, vertical offset, vertical coupling, and the probe attenuation parameters.

With the `Configure Chan Characteristics` function, you can adjust the input impedance and the and the max input frequency parameters.

Related Topics

[Configuring the Horizontal Settings](#)

Using Configure Vertical

The `Configure Vertical` function is necessary in almost every program because it allows you to enable a channel. By default, all channels are disabled. To acquire data, you must enable a channel either with `Configure Vertical` or a `Read` function. As a convenience, `Read` functions automatically enable the channels passed into them.

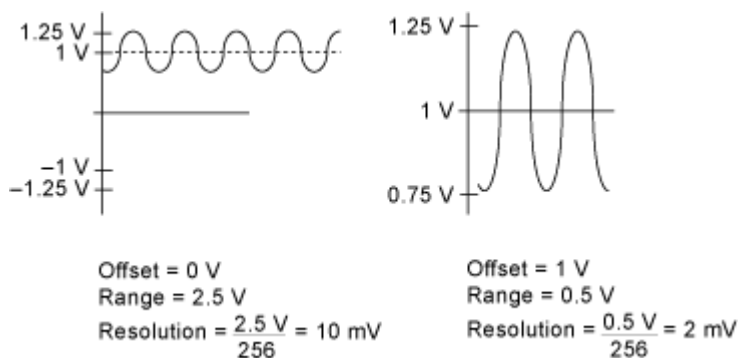
Vertical Range

`Configure Vertical` adjusts the vertical range for the digitizer. The vertical range is the full-scale (peak-to-peak) voltage range at the probe or the input to the digitizer. For example, a 10 V vertical range means the digitizer can measure a signal between -5 V and 5 V. The signal may be clipped if it exceeds this range, which means the analog-to-digital converter (ADC) is saturated, so all values above 5 V or

below -5 V are mapped to approximately 5 V or -5 V , respectively. For optimum resolution, choose the smallest vertical range that completely covers your input signal and avoids clipping. For example, if your signal is $2\text{ V}_{\text{pk-pk}}$ and your vertical range is $10\text{ V}_{\text{pk-pk}}$, you are only using one-fifth of the range of the ADC. With an 8-bit digitizer, you have 256 unique voltage levels, but if you only use one-fifth of the range, you only have 51 unique voltage levels. This results in a noisy signal from the quantization of the ADC.

Vertical Offset

The vertical offset parameter in the `Configure Vertical` function adjusts the middle of your vertical range. For example, if you set the vertical offset to 2 V with a vertical range of 10 V , your signal must be between -3 V and 7 V . Vertical offset allows you to adjust the range when analyzing signals with a DC offset. By centering the vertical offset on the DC level of your signal, you can use a smaller vertical range and optimize the resolution of your measurement.



Note Not all digitizers support using vertical offset. Refer to [Features Supported by Device](#) to determine if your digitizer supports vertical offset.

Vertical Coupling

When you need to measure a small AC signal on top of a large DC component, you can use AC-coupling by setting the vertical coupling parameter in the `Configure Vertical` function. AC-coupling rejects any DC component in your signal before it is digitized. Activating AC-coupling inserts a capacitor in series with the input.

When you change the vertical range or coupling on the digitizer, the input stage takes time to settle. When you switch from AC- to DC-coupling, the settling time is

about 0.5 ms. When you switch from DC- to AC-coupling, returned data is accurate about 20 time constants after switching to AC. A 1X probe takes about 15 ms, while a 10X probe takes 150 ms, depending on the digitizer. NI-SCOPE does not provide the delay to account for settling time; therefore, acquisitions immediately following a coupling change may yield incorrect data. However, the vertical range and coupling are set immediately when you call `Configure Vertical`, so inserting a delay in your program before calling `Initiate Acquisition` or a `Read` function allows time for the input stage to completely settle.



Note For the NI 5122, the vertical range and coupling are not set when `Configure Vertical` is called. Settling delays for a 1X probe are automatically compensated for in the driver. For other probes that require longer delays, you can call `Commit` to switch the coupling immediately. Then, you can insert the appropriate delay before calling a `Read` function or `Initiate Acquisition`.

Probe Attenuation

Probe attenuation scales your data to compensate for the attenuation of the probe. The voltage measured by the digitizer is multiplied by the probe attenuation parameter in the `Configure Vertical` function. The vertical range parameter is the voltage range you want after the probe compensation scaling. For example, if your probe attenuation is 10 and your vertical range is 10 V, the digitizer is set to measure a 1 V_{pk-pk} signal. The data returned with the `Fetch` function is 10 V_{pk-pk}.

Using Configure Chan Characteristics

Using the `Configure Chan Characteristics` function, you can configure the less common vertical parameters.

Input Impedance

The input impedance parameter allows you to switch between 50 Ω and 1 MΩ input impedance, depending on your digitizer. Refer to [Features Supported by Device](#) to find out the input impedance your digitizer supports. The allowed vertical ranges may vary depending on the input impedance. If you measure a signal from a 50 Ω output function generator, set the digitizer for 50 Ω input to match your source. Impedance matching becomes much more important with higher frequency signals

to avoid reflections of the signal that may distort your measurements. Alternatively, if you probe a circuit, it is best to use the high impedance setting on the digitizer to avoid changing the characteristics of the circuit.

To protect the 50 Ω resistor, a thermal sensing circuit opens the input if the power dissipation is too high. If this occurs, a warning is returned from all Read, Fetch, and Status functions. The circuit tries to reset itself when Read, Fetch, or Status functions are called, except on [SMC-based devices](#), which require you to reset the circuit by initiating a new acquisition. Some digitizers also support the detection of an ADC overload. ADC overload detection works the same as the 50 Ω overload detection circuit described previously.

Max Input Frequency

The max input frequency parameter sets the -3 dB cutoff frequency for a hardware analog filter. Setting this parameter to zero uses the default bandwidth of the digitizer. The filter attenuates signals greater than the cutoff frequency, which is useful for minimizing high-frequency noise when sampling at lower rates. For example, if you sample at 100 MS/s, you can resolve frequencies up to 50 MHz according to the [Nyquist theorem](#). Any noise in the signal above 50 MHz, such as harmonics of the input signal, are aliased onto a frequency below 50 MHz. The solution is to filter this noise before the signal is digitized. Refer to [Features Supported by Device](#) to find out the valid -3 dB bandwidth your digitizer supports. This hardware filter is not available in all digitizers.

Configuring the Horizontal Settings

The horizontal timing parameters apply to all enabled channels of the digitizer, which means that both channels of a two-channel digitizer sample the same amount of data at the same rate. Some digitizers support [acquiring multirecords](#). multirecord acquisitions allow you to acquire multiple, triggered waveforms very quickly. Because the triggering and timing are done in hardware, multirecord acquisitions eliminate the possibility of the operating system interrupting your program and causing the digitizer to miss triggers. To find out if your digitizer supports multirecord acquisitions, refer to [Features Supported by Device](#). The horizontal settings include the num records, sample rate, min record length, and reference position parameters.

Num Records

The `num records` parameter in the `Configure Horizontal Timing` function allows you to configure a multirecord acquisition. For a single record acquisition, set this parameter to 1.

Sample Rate

The `sample rate` parameter is the frequency at which digitized samples are stored, expressed in samples per second. This parameter is rounded up to the next legal sampling rate that your device supports. If the `enforce realtime` parameter is set to `TRUE`, the sampling rate must be set lower than the maximum real-time sampling rate of the digitizer, so data can be digitized at the requested rate. If you set the `enforce realtime` parameter to `FALSE`, NI-SCOPE allows you to specify sampling rates higher than the maximum real-time sampling rate of the digitizer. When you do this, NI-SCOPE enters [random interleaved sampling \(RIS\)](#) mode, where it acquires multiple waveforms at the maximum real-time sampling rate and reconstructs a periodic waveform.

Min Record Length

The `min record length` parameter is the minimum number of samples to store for each record in the acquisition. This parameter can also be rounded up. NI-SCOPE maintains a constant time per record, which is the `min record length` divided by the requested sampling rate. Because the sampling rate is rounded up to a legal value, the `min record length` can also be rounded up. You can fetch this resulting actual record length by calling `Actual Record Length`. Similarly, you can call `Sample Rate` to find the true sample rate used or `Actual Sample Mode` to determine if NI-SCOPE is using RIS or real-time sampling.

Reference Position

The `reference position` parameter determines the number of pretrigger versus posttrigger points that are stored. Reference position is expressed as a percent of the record, from 0 to 100. For example, a 0% reference position means that you have

the actual record length points stored after the trigger occurs, while 100% reference position means that all the samples are stored before the trigger.

Related Topics

[Configuring the Vertical Settings](#)

Configuring Triggers

NI-SCOPE provides several types of triggering options, including [immediate](#), [software](#), [hysteresis](#), [edge](#), [window](#), [video](#), and [digital](#) triggering. Each kind of triggering uses a different NI-SCOPE Configure Trigger function.

You can also use the Configure Trigger functions to set certain trigger parameters. Refer to [Trigger Parameters](#) for more information on some of the common parameters.

Attributes and Attribute Functions

Attributes—which are also called properties in LabVIEW—serve as a base for parameters. For example, the Minimum Number Points parameter in the Configure Horizontal Timing function is actually built on the Horizontal Minimum Number of Points, Horizontal Minimum Sample Rate, Horizontal Number of Records, and Horizontal Record Reference Position attributes. Because attributes and attribute functions require additional code and include limited error checking, avoid using attributes except in the following cases:

- Use attributes if NI-SCOPE does not have a high-level function that sets an attribute that you need.
- Use attributes if you need to change one variable at a time. Attributes allow you to change one variable at a time, whereas functions and parameters do not. Because NI-SCOPE efficiently caches previous settings, there is no performance penalty if you use a function when only one variable is changed at a time.

Attributes are used frequently when making waveform measurements.

Accessing Attributes

In LabVIEW, you can find attributes in the NI-SCOPE property node by dropping the property node from the NI-SCOPE palette onto your block diagram.

In C and Visual Basic, you can access attributes with the Set Attribute and Get Attribute functions.

Setting Attributes Before Reading Attributes

Attributes—which are also called properties in LabVIEW—are modified when you set them or when you call a configuration VI or function that sets them. It is important to set the attributes or call any configuration VIs or functions before reading back any attribute values for the following reasons:

- Values read are coerced depending on the current configuration of the session. If you read an attribute value and then set other attributes, the value read may no longer be valid.
- The driver verifies that the configuration of the device is valid at the time the attribute is read. It is possible to get an error when reading an attribute if the configuration is not valid at that point, even when a setting later could make it valid.
- Reading attributes causes the driver to verify the current configuration. If you change some of the settings later, those settings need to be validated again.



Note Perform all module configuration before writing data on output devices.

Utility Functions

The Utility functions perform various tasks such as resetting the digitizer and returning the revision numbers of NI-SCOPE and the instrument firmware. These functions can also help you start or stop the output of a square wave for [probe compensation](#).

Fetching Data

Fetching data involves the following main steps:

1. Declaring a waveform array (except in LabVIEW)
2. Initiating an acquisition
3. Waiting for the data acquisition to complete
4. Retrieving data from the digitizer to your host computer

Advanced Fetching Options

NI-SCOPE supports several attributes for advanced fetching operations. The two following attributes are general purpose, allowing you to fetch only a portion of the acquired waveforms.

- The `Fetch Offset` attribute is the offset in samples from which to start fetching samples. It may be negative.
- The `Fetch Relative To` attribute specifies what the retrieval offset is relative to. All digitizers support fetching relative to pretrigger and trigger. The pretrigger sample is the first pretrigger point requested with the `Configure Horizontal Timing` function. This is the typical (and default) method because it allows fetching the exact data requested with the `Configure Horizontal Timing` function. Fetching relative to the trigger means the first posttrigger sample is the first one fetched. Digitizers supporting continuous acquisition have several other options for the `Fetch Relative To` attribute.

Declaring a Waveform Array

If you are programming in C, C++, or Visual Basic, you need to declare a waveform array in your program. This array allocates space for the data that is acquired with a `Fetch` function. LabVIEW users do not need to declare a waveform because the `Fetch` call allocates it for you.

NI-SCOPE coerces up the min record length specified with the `Configure Horizontal Timing` function. You can retrieve the actual number of samples acquired by the digitizer by calling `niScope Actual Record Length`.

NI-SCOPE provides the `Actual Num Wfms` function when declaring your waveform array. This function returns the number of waveforms that are available for fetching, according to the formula:

$$\text{numWaveforms} = \text{NR} \times \text{NC} \times \text{AT}$$

where NR is the number of records, NC is the number of channels, and AT is the number of waveforms for the current acquisition type. AT is equal to 1. Using the `Actual Num Wfms` function allows your program to handle switching between different acquisition types (such as normal or flexible resolution), channel lists, and minimum record lengths without altering the fetching code.

The waveform array is a single dimension array with a size equal to the number of waveforms to fetch times the number of points to fetch in each waveform. For example, if you are fetching scaled voltage data in a C program, your code may look like the following:

```
ViReal64 *wfmPtr;
ViInt32 actualRecordLength, numWfms;
niScope_ActualRecordLength (vi, &actualRecordLength);
niScope_ActualNumWfms (vi, channelList, &numWfms);
wfmPtr = malloc (sizeof(ViReal64) * actualRecordLength * numWfms);
```

You also need to declare an `niScope_WfmInfo` structure to hold the relevant constants that describe each waveform. You need one structure for each fetched waveform. The syntax is as follows:

```
struct niScope_wfmInfo *wfmInfoPtr;
wfmInfoPtr = malloc (sizeof(struct niScope_wfmInfo) * numWfms);
```

Related Topics

[Fetching Data](#)

Initiating an Acquisition

`Initiate Acquisition` tells the digitizer to start acquiring data. During this function, the hardware is programmed with the configuration that you have chosen, and the digitizer begins sampling data and storing it to onboard memory. First, the digitizer samples the requested number of pretrigger points, ignoring any triggers that may occur. After the requested number of pretrigger points are stored, the digitizer waits for a trigger. While waiting, it continues to sample and store data into the circular, onboard memory. The trigger signals the digitizer to sample the exact number of posttrigger samples that you requested. After the posttrigger points are stored, the digitizer either stops sampling data or restarts this process for the next record. The digitizer stores the memory location of the first posttrigger sample, which allows it to calculate where the first pretrigger sample is located in memory when you fetch the waveform.

Related Topics

[Fetching Data](#)

Waiting for the Data Acquisition to Complete

After the acquisition is [initiated](#), you typically use a Fetch function with a positive timeout parameter to sleep while the digitizer acquires the data. The Fetch function polls the digitizer while waiting for the data to be acquired, and it returns the requested data when the data is available. If the data is not acquired within the time specified with the timeout parameter, NI-SCOPE returns an error.

Alternatively, you can use the `Acquisition Status` function, or the `Records Done` or `Points Done` attributes to determine when the data is available. Using these low-level functions allows you to perform other operations while the digitizer is busy. However, using a Fetch function to wait for the data is easier.

Digitizers that support continuous acquisition allow fetching while the acquisition is still in progress. When you specify a positive timeout, the Fetch function only waits for the requested data, not the entire acquisition. Furthermore, digitizers supporting continuous acquisition allow you to call a Fetch function with a timeout value of zero, which fetches all the currently available data without waiting. In this case, the

actual number of samples fetched from the digitizer is returned in the `wfmInfo` structure. For more information, refer to [Acquiring Data Continuously](#).

NI-SCOPE supports one high-level `Read` function that combines the `Initiate Acquisition` and a `Fetch` function into one call. This function is suitable for most applications, but it does not support fetching binary (non-scaled) waveforms, software triggering, or continuous acquisition. You can use `Read` to acquire and fetch some data, followed by a call to a `Fetch` function to retrieve more data.

If you want to stop the digitizer before it finishes, use `Abort`. If the digitizer supports continuous acquisition, you can fetch all the data that was acquired before you called `Abort`. However, if the trigger has not occurred in the record that the digitizer is currently acquiring, the trigger point is invalid.

Related Topics

[Fetching Data](#)

Retrieving Data from the Digitizer to the Host Computer

Fetching data refers to the process of transferring the acquired waveform from the digitizer memory to the host computer memory. This is generally done with direct memory access (DMA), which copies the binary data from the digitizer extremely quickly. Usually, the binary data is scaled to voltage during the fetch and stored in as a 64-bit floating point number. One or more waveform arrays are returned from each `Fetch` function, which allow you to display, analyze, or store the acquired data.

NI-SCOPE offers one method to fetch many types of data. This means that one function can fetch normal acquisitions, multirecord acquisitions, and continuous fetching acquisitions. For C and Visual Basic, there are four `Fetch` functions, one for each data type that may be returned. For LabVIEW, there are several `Fetch` functions that allow fetching either one or several waveforms with either a waveform cluster output or a two-dimensional array output for each data type. In LabVIEW 7.0 and later, data can also be returned as a waveform data type.

The `Fetch` functions all take a comma-delimited list of channels to fetch, and the number of samples (`numSamples`) to fetch for each waveform that is returned. If the acquisition finishes and the specified `numSamples` have not been acquired, the

Fetch functions return all the available data. The `waveformInfo` output lists the actual number of samples fetched. In LabVIEW, setting the `numSamples` parameter to `-1` fetches the actual record length, specified when you configure the acquisition. Otherwise, you can determine the actual record length by calling `Actual Record Length`.

Fetch functions may return multiple waveforms, depending on the number of channels in the channel list, the number of records, and the acquisition type. In C and Visual Basic, there is a pointer to a one-dimensional array that contains all the waveforms sequentially, where the first waveform starts at the zeroth index, the second waveform starts at the `numSamples` index, the third waveform starts at two times the `numSamples` index, and so on.

In LabVIEW, NI-SCOPE supports single waveform versions and multiple waveform versions. The cluster versions output either one cluster containing timing information and the waveform or an array of clusters. The array versions output either a single array with one waveform or a two-dimensional array with multiple waveforms.

The order of the returned data follows these rules:

- If the acquisition type returns two waveforms, they are directly next to each other.
- If multiple channels are specified, the channel's data is returned in the order of the list.
- If multiple records are specified, all Record 0 waveforms are returned before any Record 1 waveforms.

Refer to the [NI-SCOPE Function Reference Help](#) or the [NI-SCOPE LabVIEW Reference](#) for more information on Read and Fetch functions.

The most common situation is fetching two channels of data during a normal, single-record acquisition. In this case, the two waveforms are ordered the same as the channel list. Refer to [Making Multiple-Record Acquisitions](#) for more information about fetching data in multiple-record situations. Remember, you can always call a Fetch function repeatedly to fetch each waveform separately. In addition to returning waveforms, Fetch functions return an array of `wfmInfo` structures—one for

each waveform returned. The structure contains information about the waveform, including timing and scaling information.

The `waveformInfo` structure contains the necessary timing information for plotting and analyzing an acquired waveform. The `x` increment is the time, in seconds, between two samples. The relative initial `x` is the time, in seconds, of the first point in the waveform with respect to the trigger. If you acquire pretrigger samples, this value is negative. The trigger time is very accurate and has much higher resolution than the sample period because it uses a time-to-digital conversion (TDC) circuit. Using the relative initial `x` scale, the trigger always occurs at time equals zero. In LabVIEW, the Fetch functions that return a cluster contain the relative initial `x` value and the `x` increment. Wiring that cluster directly to the graph plots the waveform versus time. The absolute initial `x` parameter is only supported on devices with continuous acquisition. Refer to [Acquiring Data Continuously](#) for more information on continuous acquisition.

The `waveformInfo` structure also contains the gain and offset scaling factors that allow you to convert binary data to voltage. Often, you need to maximize the speed of the application while acquiring data. Fetching binary data saves time because it avoids the scaling operation and uses significantly less memory (on an 8-bit device, 1 byte instead of 8 bytes per sample). However, you may want to scale the binary values to a voltage at a later time. To do this, use the gain factor and offset values from the `waveformInfo` structure in the following formula:

$$\text{Voltage Value} = \text{Gain} \times \text{Binary Value} + \text{Offset}$$

Related Topics

[Fetching Data](#)

Using Fetch VIs

What Are Fetch Acquisition VIs?

Unlike Read VIs, Fetch VIs such as [niScope Fetch Poly](#) and [niScope Fetch Measurement Poly](#)—acquire data asynchronously. That means these VIs store data on your digitizer until you need it, which frees up your computer to perform other

tasks. However, fetching requires additional VIs that initiate waveform acquisitions and query your device to verify if the acquisition is complete.

Using Fetch Acquisition VIs

To fetch data, write a program that follows these general steps:

1. Create a new instrument session using [niScope Initialize](#).
2. Configure your digitizer with NI-SCOPE Configuration VIs.
3. Call [niScope Actual Record Length](#) to find the actual number of points your digitizer acquires for each channel.
4. Call [niScope Initiate Acquisition](#) to initiate a waveform acquisition. Data will be acquired simultaneously on all enabled channels.
5. Call [niScope Acquisition Status](#) in a loop to check if the acquisition is complete or in progress. This step is optional and useful if you want to perform other tasks during the acquisition. The Fetch VIs also accept a timeout that can make NI-SCOPE wait until the acquisition is complete.
6. Call the Fetch VI you need. You can call multiple fetches if you are acquiring data from more than one channel or measurement. You use the same Fetch VIs for simple data acquisition and advanced continuous acquisition, such as fetching multiple records. Although the VIs only have the numSamples input parameter to determine how what data is retrieved, NI-SCOPE offers several properties that allow you to specify advanced fetching options. The [Fetch Relative To](#) and [Fetch Offset](#) properties allow you to explicitly choose the portion of a record to retrieve. The [Fetch Record Number](#) and [Fetch Number of Records](#) properties allow specifying the records to retrieve during a multirecord acquisition.
7. Call [niScope Close](#) to close the current session.

Using Fetch Functions

What Are Fetch Acquisition Functions?

Unlike Read functions, Fetch functions—such as [niScope_Fetch](#) and [niScope_FetchMeasurement](#)—acquire data asynchronously. That means these functions store data on your digitizer until you need it, which frees up your computer to

perform other tasks and allows for better system performance. However, fetching requires additional functions that initiate waveform acquisitions and query your board to see if the acquisition is complete.

You use the same Fetch functions for both simple data acquisition and advanced data acquisition such as multiple-record fetching. Although the functions have only the `numSamples` parameter to determine how data is retrieved, NI-SCOPE offers several attributes that allow you to specify advanced fetching options. The `NISCOPE_ATTR_FETCH_RELATIVE_TO` and `NISCOPE_ATTR_FETCH_OFFSET` attributes allow you to explicitly choose the portion of a record to retrieve. The `NISCOPE_ATTR_FETCH_RECORD_NUMBER` and `NISCOPE_ATTR_FETCH_NUM_RECORDS` attributes allow specifying the records to retrieve during a multirecord acquisition.

Using Fetch Acquisition Functions

To fetch data, write a program that follows these general steps:

1. Create a new instrument session using `niScope_init`.
2. Configure your digitizer with NI-SCOPE Configuration functions.
3. Call `niScope_ActualRecordLength` to find the actual number of points your digitizer acquires for each channel.
4. Call `niScope_InitiateAcquisition` to initiate a waveform acquisition. Data will be acquired simultaneously on all enabled channels.
5. Call `niScope_AcquisitionStatus` in a loop to check if the acquisition is complete or in progress. This step is useful if you want to perform other tasks during the acquisition. The Fetch functions can also accept a timeout to force NI-SCOPE to wait until the acquisition is complete.
6. Call the Fetch function you need. You can call multiple fetches if you are acquiring data from more than one channel or measurement.
7. Call `niScope_close` to close the current session.

Acquiring Data Continuously

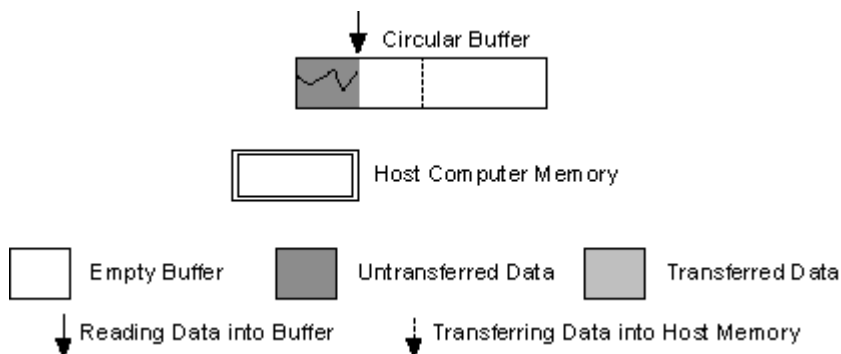
Continuous acquisition is the ability to transfer data from the digitizer to the host computer memory while the digitizer is still acquiring data, which can be useful for the following applications:

- Acquiring records larger than available memory
- Fetching triggered records while other records are being acquired
- Acquiring more records than fit in digitizer memory
- Fetching the most recent data
- Acquiring waveforms at hardware-timed intervals

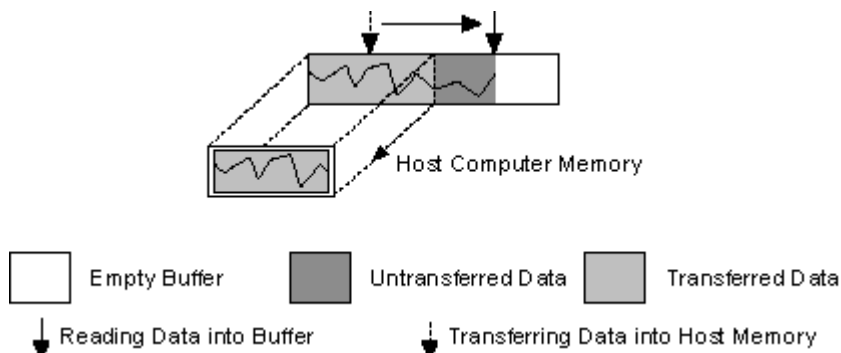
How Continuous Acquisition Works

NI digitizers contain a large amount of onboard memory, generally 8, 16, 32, or 256 MB per channel. This memory is divided into individual records when acquiring data. For example, if you are acquiring two records, 8 MB of memory is divided into two 4 MB records. Each record is treated as a circular buffer. When the digitizer starts acquiring data, the samples are placed in the beginning of the buffer. The digitizer acquires the requested number of pretrigger samples—that is, the actual record length times the reference position divided by 100. After the pretrigger samples are complete, the digitizer waits for a trigger. While waiting, the digitizer continues to acquire and store data. This data is placed in the circular buffer, so after the buffer is filled, the digitizer starts overwriting data at the beginning. If a trigger never occurs, the digitizer waits for a trigger forever. After the trigger occurs, the digitizer samples the requested number of posttrigger samples, and the record is complete. If another record is requested, the digitizer restarts the acquisition, moving to the next record in memory.

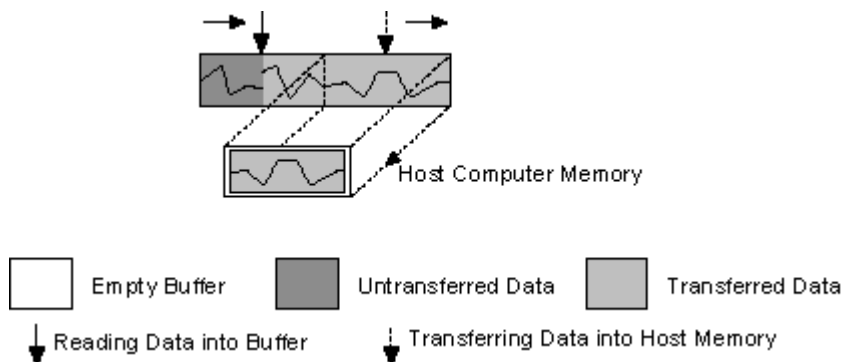
Continuous acquisition refers to the digitizer's ability to fetch data from its own circular buffer to the host computer's memory while it is acquiring data. When you call `Initiate Acquisition`, the digitizer starts acquiring data into its circular buffer, as shown in the following figure.



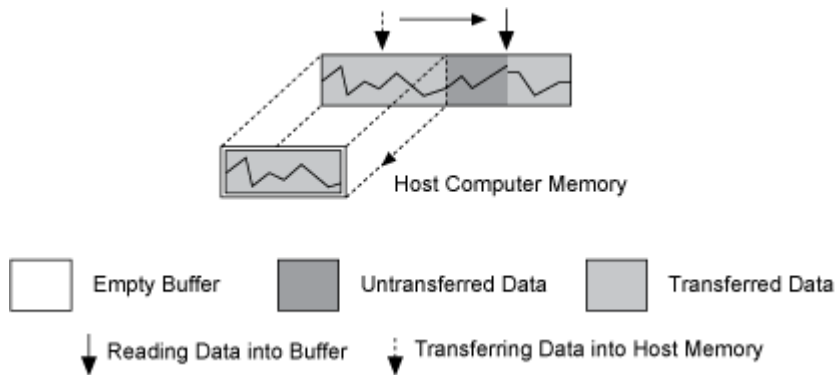
You can then use a Fetch function to send the first chunk of data to the host computer while the digitizer continues to acquire more samples, as shown in the following figure.



When the circular buffer is filled on the digitizer, it starts to overwrite the data at the beginning of the buffer, as shown in the following figure.



Ideally, you have already copied and saved the data that is being overwritten in the host computer.



If you do not send a trigger, the digitizer continues to acquire data forever or until you call `Abort`, `Reset`, or `Close`.

Fetching Continuous Acquisition Data

No special configuration is necessary for continuous acquisition programs. The only difference is how the data is fetched. The `Fetch` functions all work for continuous acquisition, but additional attributes are available to specify what data to fetch. In particular, the `Fetch Relative To` attribute supports values of `Read Pointer`, `Now`, and `Start` in addition to the standard `Trigger` and `Pretrigger` values. These values allow specifying different locations within the record from which to start fetching. When fetching data continuously, the possibility always exists that the data has been overwritten in the onboard memory before you attempt to fetch it. NI-SCOPE returns an error in this situation.

When fetching data continuously, the relative initial `x` value returned in the waveform info structure is not valid until the trigger occurs. Often, the trigger never occurs during continuous acquisitions. However, the waveform info structure also contains an absolute initial `x` value, which is a free-running [timestamp](#) counter on the digitizer. Additional status information about your acquisition is available by using the `Fetch Points Done` and `Fetch Records Done` attributes. The `points done` is the number of samples available in the record specified by the `Fetch Record Number` attribute, starting at the `Fetch Offset` attribute that is relative to the `Fetch Relative To` attribute. NI-SCOPE also supports a `Fetch Backlog` attribute, which is the number of samples that have not been fetched when performing a continuous acquisition.

Acquiring Records Larger than Available Memory

The standard use of continuous acquisition is to fetch a record that is larger than the available memory on the digitizer. Because the data is fetched as it is acquired, the digitizer memory can be overwritten. At slow sampling rates, you can fetch data forever by setting up an acquisition that is never triggered and is repeatedly fetching. At faster sampling rates, the host computer may not be able to fetch as fast as the digitizer samples data. If the data that is being fetched is overwritten, NI-SCOPE returns an error message from the Fetch function. Look at the `Fetch in Chunks` example to see how a waveform can be reconstructed with the data from multiple fetch calls. Look at the `Fetch Forever` example to benchmark how much data you can acquire at a given sampling rate before the data is overwritten.

To fetch a record that is larger than memory, set the `Fetch Relative To` attribute to `Read Pointer`. This positions the beginning of the fetch operation at the start of the record when you initiate a new acquisition. After every fetch, the read pointer is incremented to be the sample after the last sample retrieved. Therefore, you can repeatedly fetch relative to the read pointer, with a retrieval offset of zero, to acquire a single, infinite record.

If you specify a positive timeout with a Fetch function, it waits for the requested number of samples. Alternatively, specifying a timeout of zero acquires the number of samples currently available (up to a maximum of the `numSamples` parameter). The waveform info structure returns the actual number of samples fetched. Using a timeout of zero achieves slightly better performance because the status of the digitizer is queried less often.

A separate read pointer is stored for each channel, so you can alternate fetching different channels. The read pointer is also reset to zero when you fetch from a different record.

Fetching Triggered Records while Other Records Are Being Acquired

Fetching records continuously can greatly speed up applications that have a very slow trigger rate. If you are acquiring at 100 MS/s, but you only receive a trigger once every second, you can completely fetch each record as soon as it is acquired by fetching each record individually. You can set the `Fetch Num Records` attribute to 1 and increment the `Fetch Record Number` attribute for each record you

want to fetch. Then, using a positive timeout with any of the Fetch functions causes the digitizer to wait only for the next record to be acquired before it returns the waveform.

For lower-level control, NI-SCOPE supports the `Records Done` attribute that you can poll to determine the current status of your acquisition.

Acquiring More Records Than Fit in Digitizer Memory

During typical multirecord operation, each buffer in memory must be large enough to contain all the pretrigger and posttrigger samples. If you request 1,000 points with `Configure Horizontal Timing`, each record in memory must be at least 1,000 points; otherwise NI-SCOPE returns an error. Typically, all the records are required to fit in memory, so you can wait until the entire acquisition is finished before fetching anything.

However, in certain cases, you may want to acquire more records than would fit in the onboard memory. In this situation, the records in memory are also circular. That means if you configure four records and only three fit in memory, the fourth record overwrites the first record. Ideally, you have time to fetch the first record to the host computer memory before it is overwritten.

To fetch more records than fit in memory, set the Boolean attribute `Enable More Records Than Memory` to `TRUE` and configure the desired number of records with the `Configure Horizontal Timing` function. You must fetch the records individually as described in [Fetching Triggered Records while other Records are Being Acquired](#). If the record you are attempting to fetch has been overwritten, NI-SCOPE returns an error.



Note The total number of records that you can acquire is limited. Each record requires up to 64 bytes of page-locked memory. The actual amount of page-locked memory depends on the amount of physical memory and the number of other devices being used in your host computer. You can configure as many records as you need, but save your work first.

Fetching the Most Recent Data

Another application of continuous acquisition is fetching a few of the most recent data points. This technique is useful if you want chunks of nontriggered data with the same configuration parameters. It avoids the software overhead necessary to

reconfigure the acquisition, and your program never needs to wait for the data to be acquired.

To fetch the most recent data, initiate an infinite acquisition—one that never triggers—by configuring software triggering. When you want the most recent data points, call a Fetch function with the `Fetch Relative To` attribute set to `now`, which specifies that the offset is from the last sample acquired by the digitizer. You must set the `Fetch Offset` attribute to the negated number of samples you want.

Acquiring Waveforms at Hardware-Timed Intervals

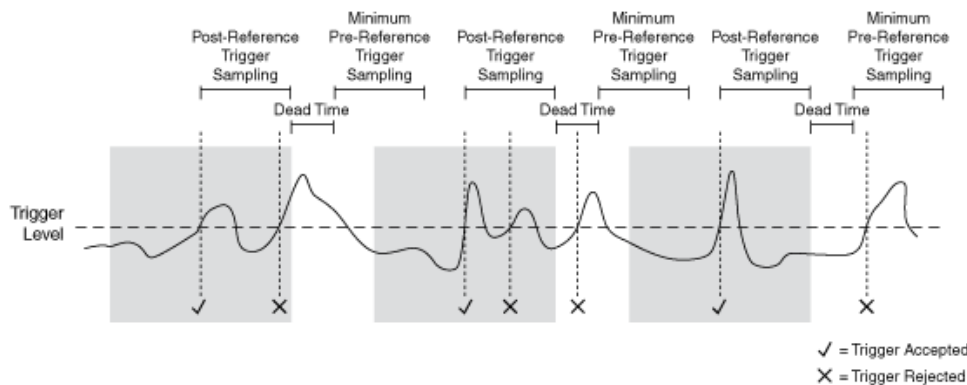
Setting the `Fetch Relative To` attribute to `Start` sets the starting fetch position to be the first sample acquired by the digitizer. If the trigger happens immediately, the first point sampled by the digitizer would equal the first pretrigger point. However, this is generally not true because the digitizer usually has to wait for a trigger. While the digitizer waits, it continues to sample data, possibly forever. As it samples data into the circular, onboard memory, the original data is eventually overwritten. As in other cases, if you attempt to fetch data that is overwritten, NI-SCOPE returns an error. A typical use of fetching relative to start is to acquire nontriggered waveforms at precise intervals. You can use the digitizer sample clock to precisely time the duration between waveforms. Suppose you want 500 points every millisecond, while sampling at 100 MS/s. You can set the `Fetch Relative To` attribute to start at the beginning of your program. Then, for every iteration of a loop, call a Fetch function to retrieve 500 samples, and increment the `Fetch Offset` attribute by 100,000 samples. Because 100,000 samples are acquired every millisecond when sampling at 100 MS/s, this program is effectively using the sample clock to precisely time the interval between acquired waveforms.

Making Multiple-Record Acquisitions

Some NI digitizers support multiple-record acquisitions, also known as retriggerable acquisitions, which allow you to capture multiple, triggered waveforms without software intervention. NI-SCOPE stores each record in separate memory locations on the digitizer. Refer to [Features Supported by Device](#) for a listing of digitizers that support multiple-record acquisitions.

The main benefit of multiple-record acquisitions is that you can acquire numerous triggered waveforms quickly. Multiple-record acquisitions allow hardware rearming of the digitizer. The digitizer hardware must transition between several states (given by the digitizer's Acquisition Engine State Diagram) before it is ready to accept a subsequent trigger. If two triggers arrive, one shortly after the other, the digitizer may or may not detect the second trigger.

- If the second trigger arrives before the first record completes, it will not be accepted. Only one trigger can be accepted in any one record in the NI-SCOPE driver.
- If the second trigger arrives after the first record completes, but before the dead time has elapsed, it will not be accepted.
- If the second trigger arrives after the first record completes and after the dead time has elapsed, but before the minimum pre-reference trigger sampling for the subsequent record is complete, it will not be accepted.
- If the second trigger arrives after the first record completes, the dead time has elapsed, and the minimum pre-reference trigger sampling is finished, the trigger will be accepted.



Note The information above assumes that there are no holdoff values set (for example, trigger holdoff).

Minimum Time Between Reference Triggers

The minimum time between triggers can be calculated with the dead time and the time per record as follows:

Minimum Time Between Triggers = Time Per Record + Dead Time

The maximum trigger rate can be calculated as follows:

Maximum Trigger Rate (Hz) = 1/(Minimum Time Between Triggers)

When the reference trigger type is set to immediate and all holdoff values = 0 seconds, the digitizer will trigger at the maximum trigger rate regardless of any external signal.

Some digitizers specify a minimum rearm time. Minimum rearm time is the minimum time between reference triggers as the record length approaches a minimum (for example, record length = 1 sample). Therefore, when the record length is 1 sample, Maximum Trigger Rate (Hz) = 1/(minimum rearm time).

Multiple-Record Example

For an introduction to multiple-record acquisitions, refer to the `Multi Record` example. Modify the `niScope_ConfigureHorizontalTiming` function in your application, setting the `numRecords` parameter to the number of records you want to acquire. The digitizer acquires an additional record each time a trigger is accepted until all the requested records have been acquired.

Related topics

[Trigger Holdoff Fundamentals](#)

Fetching Multiple-Record Acquisitions

You use the same Fetch functions discussed in [Acquisition Functions—Reading versus Fetching](#) for retrieving multirecord acquisitions. However, two additional attributes are available for specifying which records to retrieve. The `Fetch Record Number` attribute is the zero-based index of the first record to fetch, and the `Fetch Number of Records` attribute specifies how many records to fetch. By default, the `Fetch Number of Records` is -1, which means fetch all the records starting at the `Fetch Record Number`.

The Fetch functions wait for specified number of samples to be acquired when the timeout parameter is positive. During a multirecord acquisition, they wait for the requested number of samples in each record. Because the number of records attribute defaults to -1 (or all the records), Fetch functions wait for all the specified number of samples in all the records.

Fetching multiple records with a single Fetch function requires understanding the order of the returned waveforms. All record 0 waveforms come before all record 1 waveforms. For example, fetching data with a channel list of "0,1" for three records results in the following order:

- Channel 0 Record 0
- Channel 1 Record 0
- Channel 0 Record 1
- Channel 1 Record 1
- Channel 0 Record 2
- Channel 1 Record 2

For C and Visual Basic users, the waveforms are all packed into a one-dimensional array that is declared using code such as the following:

```
// Set numWfms to 6, because the acquisition is f
or 2 channels times 3 records.
niScope_ActualNumWfms (vi, "0,1", &numWfms);

// Fetch the coerced record length
niScope_ActualRecordLength (vi, &actualRecordLen
gh);

// Declare memory for the waveforms and waveform
info structs
wfm = malloc (sizeof (ViReal64) * actualRecordLen
gth * numWfms);
wfmInfo = malloc (sizeof (struct niScope_wfmInfo)
* numWfms);
```

The first waveform starts at `wfm[0]`, the second waveform at `wfm[actualRecordLength]`, and so on.

For LabVIEW users, the waveforms are returned in either a two-dimensional array or an array of clusters that include timing information. In both cases, you can use the index array function to extract the waveform of interest.

You can also fetch each record individually by setting the `Fetch Number of Records` parameter to 1. Then, in a loop set the `Fetch Record Number` attribute to the zero-based index of the record you want to fetch and call one of the `Fetch` functions.

Making Waveform Measurements

This section describes the different ways to make waveform measurements, which are processed in software by NI-SCOPE. For more information, refer to the following:

[Fetching Scalar and Array Measurements](#)

[Making Scalar Measurements](#)

[Making Array Measurements](#)


Processing Time Domain Data

Often the time-domain data returned from a digitizer requires processing before measurements are done. To do this, register an array measurement as a processing step by using the `AddWaveformProcessing` function. At the time of registering the processing step, the entire set of `NISCOPE_ATTR_MEAS` parameters are cached. Then when a measurement function is called, the processing measurements are completed with this cached set of parameters, and the result is used in the measurement. This allows streaming together of measurements. For example, if you register `Add Channels` and `Bessel Filter` as processing steps, fetching a frequency measurement actually gives you the frequency of the filtered summation of two channels.

Fetching Scalar and Array Measurements

You can fetch scalar and array measurements with the `Fetch Measurement`, `Fetch Measurement Stats`, and `Fetch Array Measurement` functions. The

channel list and timeout parameters are identical to the waveform functions. However, instead of returning waveforms, these functions return either the specified scalar measurement or the specified array measurement. In LabVIEW, versions exist that fetch either a single measurement or an array of measurements. The C and Visual Basic versions return an array of measurement results, just as the Fetch functions can return an array of waveforms. The order of the returned results is the same as in the Fetch functions. C and Visual Basic users can dynamically declare memory for the measurement results using code such as the following.

 **Note** The following sample code uses the `Actual Meas Wfm Size` function to query how many samples are available in the resulting array measurement.

```
ViConstString channelList = "0,1";
ViReal64 *results;
niScope_ActualNumWfms (vi, channelList, &numWfms);
results = malloc (sizeof (ViReal64) * numWfms);
niScope_FetchMeasurement (vi, channelList, timeout, NISCOPE_VAL_RISE_TIME, results);
ViReal64 *measWfm;
struct niScope_wfmInfo *measWfmInfo;
niScope_ActualMeasWfmSize (vi, NISCOPE_VAL_FFT_AMP_SPECTRUM_DB, &measWfmSize);
measWfm = malloc (sizeof (ViReal64) * measWfmSize * numWfms);
measWfmInfo = malloc (sizeof (struct niScope_wfmInfo) * numWfms);
niScope_FetchArrayMeasurement (vi, channelList, timeout, NISCOPE_VAL_FFT_AMP_SPECTRUM_DB, measWfmSize, measWfm, measWfmInfo);
```

Steps for Fetching Scalar and Array Data with Fetch Functions

1. Create a new instrument session using the [niScope Initialize VI](#) or the [niScope_init](#) function.
2. Configure the digitizer with NI-SCOPE Configuration functions.

3. Call the [niScope Initiate Acquisition VI](#) or the [niScope_InitiateAcquisition](#) function to initiate a waveform acquisition. Data is acquired simultaneously on all enabled channels.
4. Call the [niScope Acquisition Status VI](#) or the [niScope_AcquisitionStatus](#) function in a loop to verify whether the acquisition is complete or in progress. Alternatively, you can pass a nonzero timeout value to the `Fetch Measurement` or the `Fetch Array Measurement` function.
5. Call a `Fetch Measurement VI` or function.

Related Topics

[Waveform Measurements](#)

[Using Attributes in Waveform Measurements](#)

[Fetching Statistics from Waveform Measurements](#)

Using Attributes in Waveform Measurements

By default, the `Fetch Measurement` functions use the entire acquired waveform for the analysis. However, all the fetching attributes apply when determining what data is used for the measurement. In addition to the normal fetch attributes such as `Fetch Relative To` and `Fetch Offset`, the `Fetch Meas Num Samples` attribute allows you to specify the number of samples fetched for performing the measurement. By default, this attribute is `-1`, which fetches the actual record length. This attribute is the same as the `numSamples` parameter in the other `Fetch` functions.

Configuring Reference Levels

Use the channel-based attributes to set the [reference levels](#). The `Measurement Library` example shows how to set the reference levels. You can customize how NI-SCOPE configures the percentages used for the reference levels with the `Percentage Method` attribute. This attribute uses any of the following constants:

- **Min Max**—This method uses the measurement's voltage minimum and voltage maximum as 0% and 100%. These algorithms find the absolute minimum and maximum in the waveform, which is useful for sine waves and triangle waves where the histogram method does not work.
- **Low High**—This method uses the measurement's voltage low and voltage high as 0% and 100%. The voltage low is the voltage of the histogram bin with the most hits below 40% of the waveform's voltage peak-to-peak value. The voltage high is the voltage of the histogram bin with the most hits above 60% of the waveform's voltage peak-to-peak value. These measurements use the last-acquisition histogram method to find the most common high and low voltages, which is useful for ignoring the preshoot and overshoot on a square wave.
- **Base Top**—The voltage base and voltage top measurements correspond to 0% and 100% with this method. These measurements use the last-acquisition histogram method if the most common histogram bin contains a substantial number of the total points. Otherwise, it returns the absolute minimum and maximum values in the waveform, which is a useful default value for most waveform types.

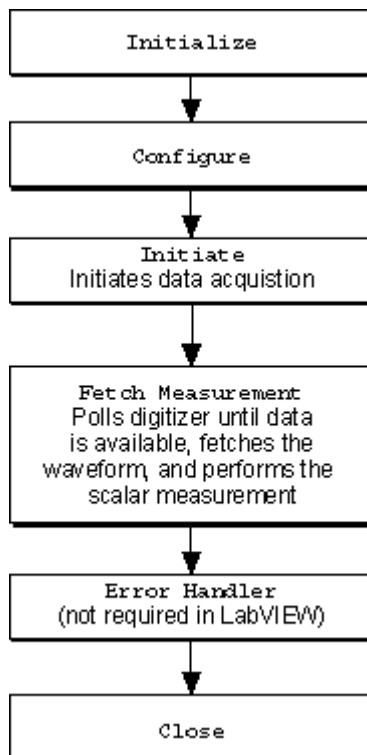
You can also configure the reference levels in terms of voltage with the Ref Level Units attribute.

Fetching Statistics from Waveform Measurements

The `Fetch Measurement Stats` function returns the current measurement result as well as statistics for this measurement over multiple acquisitions. Every time you fetch a measurement, NI-SCOPE keeps a history of the measurement values, which allows NI-SCOPE to compute the mean, standard deviation, minimum, and maximum value of each scalar measurement. To clear the statistics history, use `Clear Waveform Measurement Stats`.

Making Scalar Measurements

The following figure shows the programming flow for making scalar measurements.

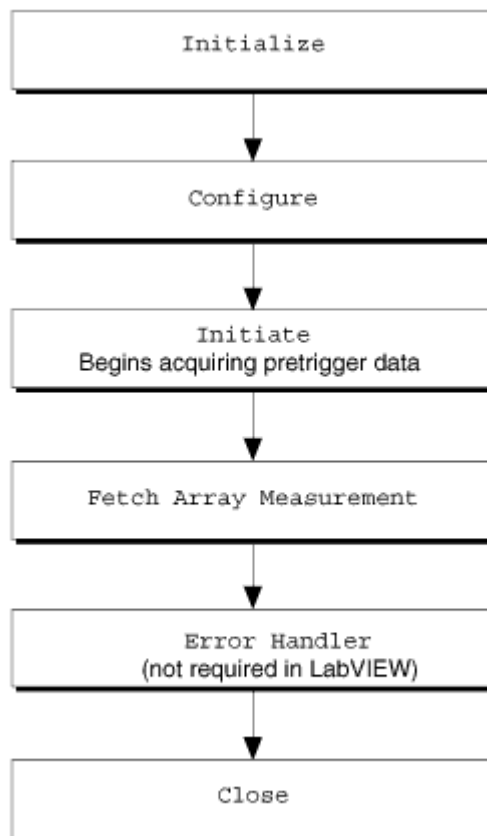


Scalar Measurement Example

Refer to the `Measurement Library` example for sample code you can use to create your own application. This example demonstrates fetching scalar measurements and statistics. The `Advanced Measurement Library` example demonstrates how to fetch a scalar measurement on a processed waveform.

Making Array Measurements

The following figure shows the programming flow for making array measurements.



Array Measurement Example

Refer to the `Advanced Waveform Measurement` example for sample code you can use to create your own application. The `Digital Filtering and Windowing` examples for LabVIEW also use array measurements.

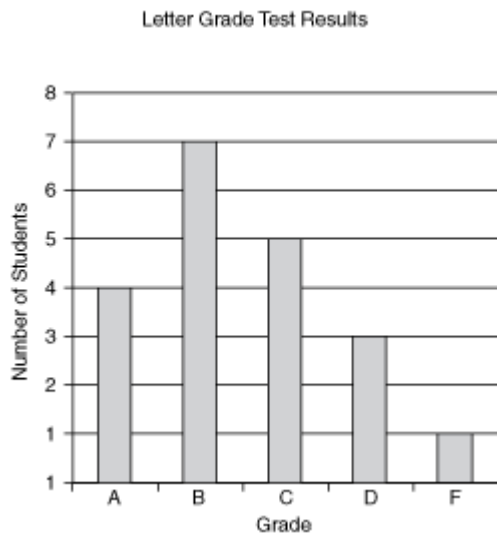
Histogram Measurements

Using histograms and histogram statistics with NI high-speed digitizers provides an effective visualization of waveform behavior such as the characterization of jitter in signals. Many applications call for signals with accurate and reliable periodicity and precise amplitude accuracy. Histograms are very useful in characterizing amplitude and temporal variations of waveforms. You can make measurements such as cycle-to-cycle and edge-to-edge jitter with time histogram measurements; you can make measurements such as pulse height analysis and distortion with voltage histograms.

In some cases the statistical distribution of response values gives richer insights than numerical representations such as mean, median, standard deviation, and mode. Histograms are specialized plots that capture statistical data about how many times specific values occur. Histograms are usually plotted with the dependent variable along the vertical axis and the independent variable along the horizontal axis.

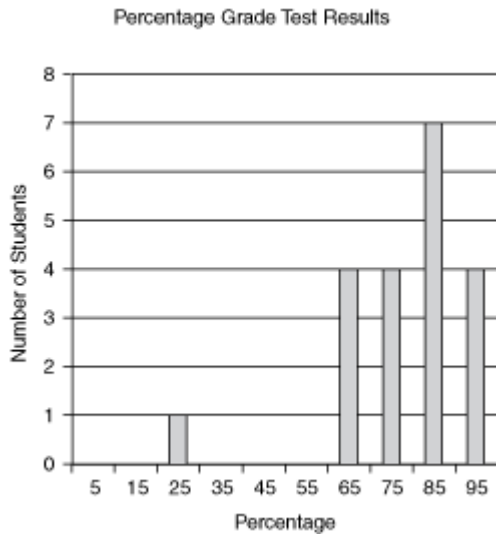
Histograms from Discrete Data

To create a histogram from discrete response data, a count is taken each time a discrete value occurs. For example, the following figure is a histogram of 20 students' letter grades on a test. The discrete values are the letter grades and the counts represent the number of students who received each grade.



Histograms from Continuous Data

To create a histogram from continuous response data (as in a frequency-distribution table), the response range is divided into ranges called bins. The value of the bin is the mean range the bin covers. Using the previous example, imagine the tests were graded with a percentage instead of a letter. The histogram in the following figure represents this distribution using bins that represent 10% increments.



Related Topics

[Time Histograms](#)

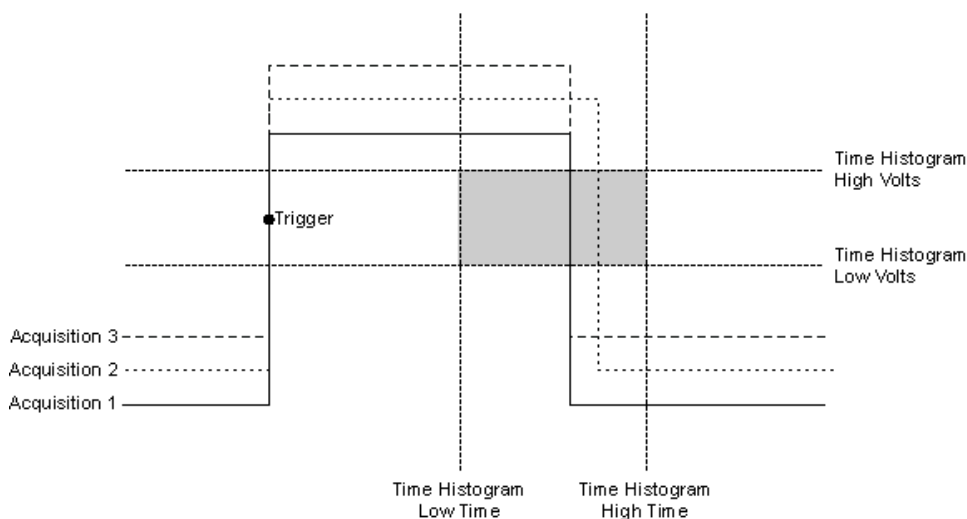
[Voltage Histograms](#)

Time Histograms

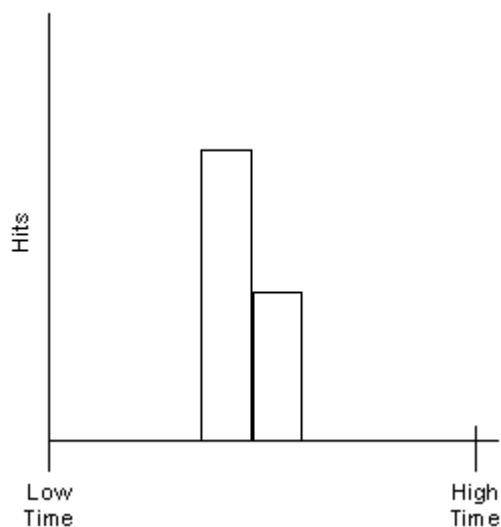
Time histograms place samples that fall within a defined voltage range and time window into bins based on their time relative to a trigger point. The time resolution of time histograms depends directly on the precision with which samples can be positioned in time relative to the trigger. More precise positioning of the sample in time allows the use of more histogram bins of smaller size, increasing histogram resolution. Often, digitizers can only resolve samples in time with a resolution equal to the sample period. Many NI digitizers, however, include a high-resolution Time-to-Digital Conversion (TDC) circuit. Using a process called time-stamping, the TDC circuit locates samples relative to the trigger point with exceptional precision. For example, the NI 5122 is equipped with a TDC circuit that allows time-stamping of samples with 100 ps resolution, while using a 10 ns sample period. Thus, TDC circuitry and time-stamping allow you to sort data into highly resolved time histogram bins and maximize histogram resolution.

The following figures demonstrate how time histograms are constructed. The first figure is a time-domain waveform sampled by a high-speed digitizer. The second

figure shows the corresponding time histogram. Multiple pulses are acquired, and an edge trigger is used to align the rising edge of each pulse. Histogram voltage (vertical) discriminator levels are configured to define a window around the falling edges of each pulse—the shaded region in the figure. Make sure you capture only the falling edges of the waveform when setting these discriminator levels—otherwise, the histogram appears perfectly uniform.



When you use this setup, every sample on the falling edge is added to the time histogram shown in the following figure. In this example, both the first and third acquisitions have a falling edge at the same time, while the second acquisition is later. The histogram captures this statistical information.



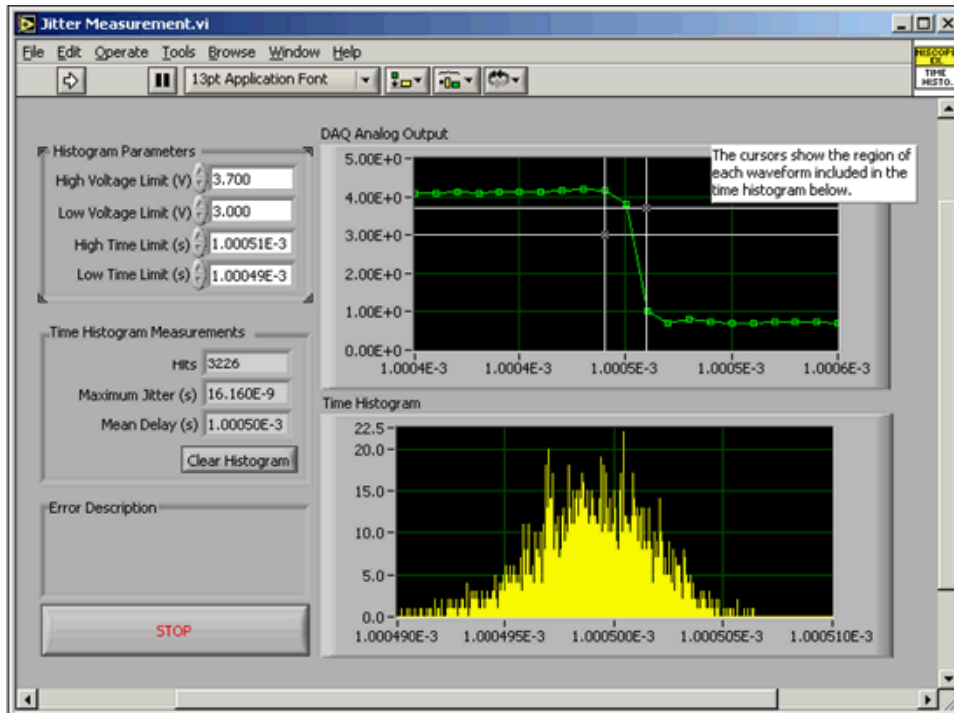
Time Histogram Measurements Application

Histograms can be used to characterize delay jitter between the input and output of a real-time I/O system. By definition, a real-time system is deterministic in time, but it is useful to know exactly how deterministic a given measurement system is. Time histogram measurements can be used to characterize the jitter in real-time systems, which in turn allows measurement of the determinism of a real-time I/O system.

Let's look at a specific example: feeding a square wave into the analog input of a real-time system while sampling at a rate of 1 kHz. The samples acquired are output to an analog channel by the real-time system one clock cycle later. This provides a deterministic delay of 1 msec between the analog input and the analog output of the real-time system. Measurement of the jitter in this delay between the input and output yields a measure of the reliability of the real-time I/O system.

In this example, the NI 5122 two-channel high-speed digitizer is used to measure the jitter. The input square wave and the analog output of the real-time system are sampled by the NI 5122. A time histogram of the delay from the edge on one channel to the same edge on the second channel is created. Statistics from the time histogram such as Time Histogram Min, Time Histogram Max, Time Histogram Hits and Time Histogram Mean yield measurements of the maximum jitter, the number of measured samples in the histogram, and the mean delay.

The results shown in the following figure yield a Gaussian distribution of the variations of the delay, meaning that this jitter is most likely derived from random electrical noise in the measurement system.



Histogram Measurements of Delay Between Analog Input and Analog Output on a Real-Time I/O System.



Note Maximum (peak-to-peak) jitter is approximately 16 ns for the Real-Time I/O system of 1 ms delay.

Related Topics

[Creating Time Histograms](#)

Creating Time Histograms

To create a [time histogram](#), you must use attributes/properties and follow these general steps:

1. Set the size of the histogram using the [Time Histogram Size](#) property or the `NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_SIZE` attribute. The default size is 256 bins. The more bins used, the greater the histogram resolution.
2. Time limits determine the temporal window. Set the time limits for the histogram using the [Time Histogram Low Time](#) property or the `NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOW_TIME` attribute. for the beginning time.

Use the [Time Histogram High Time](#) property or the `NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIGH_TIME` attribute for the ending time. Set these values during the first measurement after the histogram history is cleared.

3. Histogram voltage limits determine the vertical window. Set the voltage limits for the acquisition using the [Time Histogram Low Volts](#) property or the `NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOW_VOLTS` attribute and the [Time Histogram High Volts](#) property or the `NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIGH_VOLTS` attribute. These values can change every acquisition.



Note Together the time and voltage limits determine the discriminators for the histogram.

4. Call the [niScope Clear Waveform Measurement Stats VI](#) or the `niScope_ClearWaveformMeasurementStats` function to erase the histogram history between acquisitions.

Time Histogram Example (LabVIEW Only)

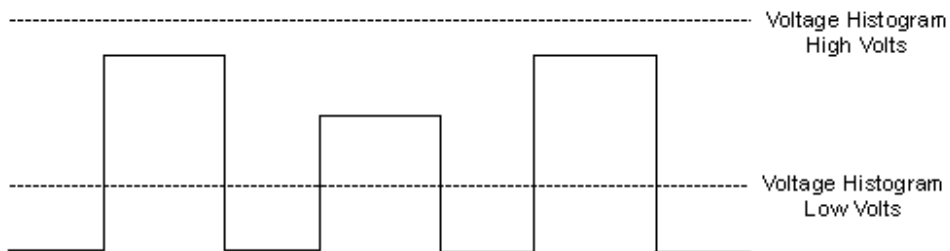
Refer to the `niScope_EX_Time_Histogram.vi` example for sample code you can use to create your own application.

Related Topics

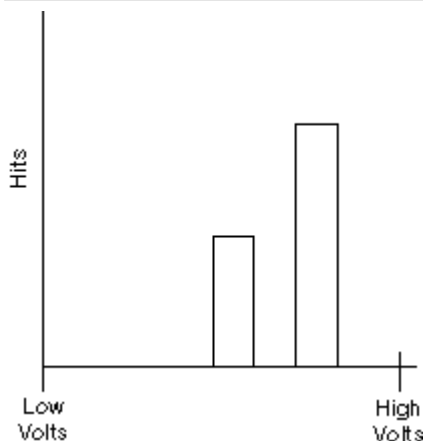
[Time Histograms](#)

Voltage Histograms

Many scientific and engineering fields require characterization of statistical variations in signal amplitude. Pulse height analysis, a common application in many scientific disciplines, can benefit from voltage histograms. In electronic measurements, signal distortion and quality can also be characterized by voltage histograms. Voltage histograms eliminate the time information from multiple acquisitions and place each point in the acquired waveform into a voltage bin. This is useful for analyzing the statistical amplitude variations of signals. The following figure shows a time-domain waveform sampled by a high-speed digitizer. The second figure shows the corresponding voltage histogram.



 Note The higher amplitude pulse occurs twice. The lower amplitude pulse occurs only once.



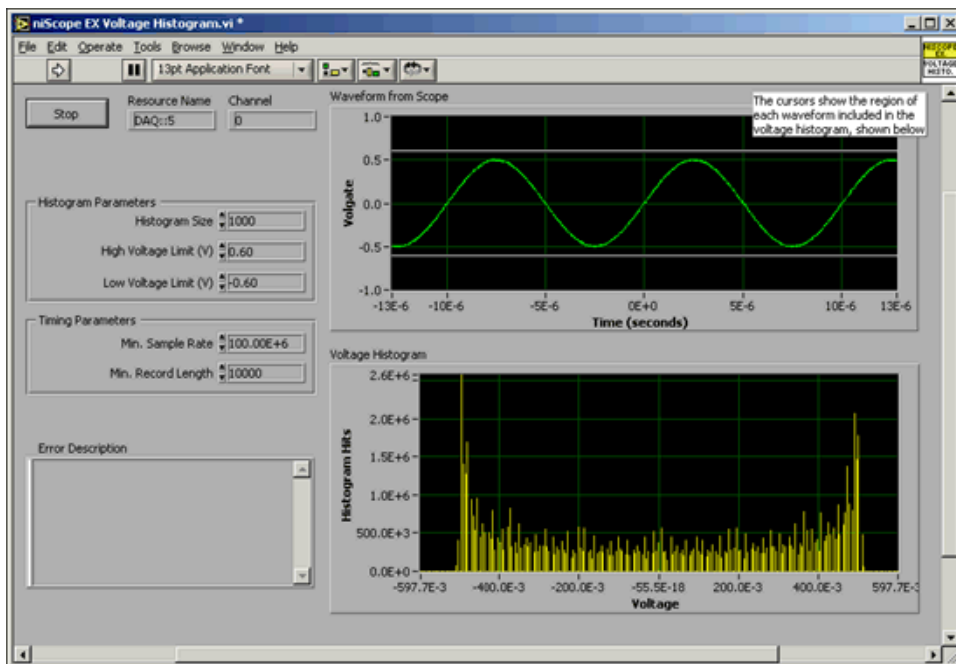
Voltage Histogram Measurements Application

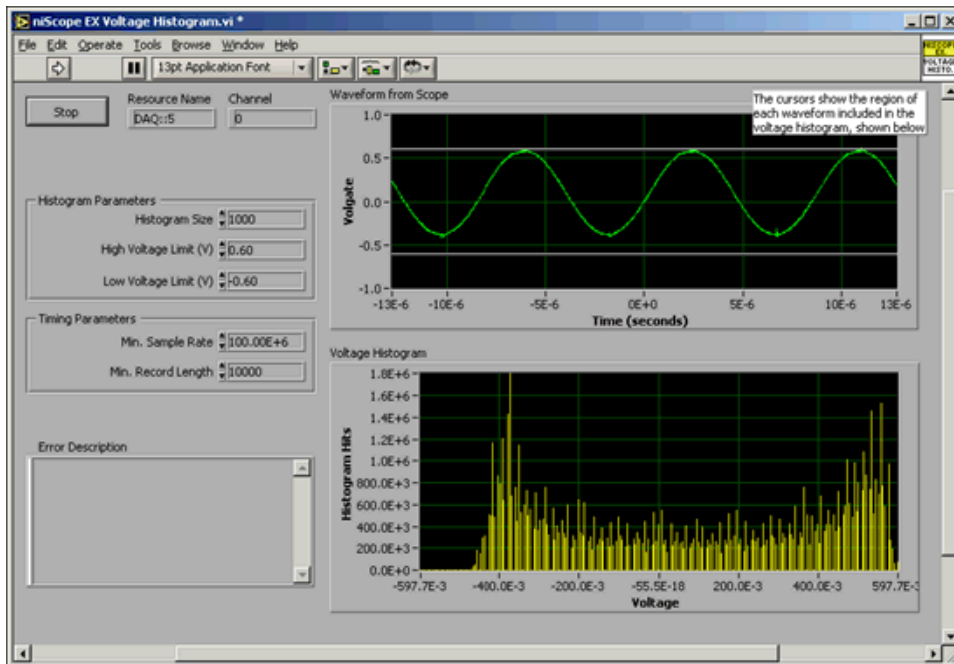
One application calling for voltage histogram measurements is electronic signal characterization. Signal sources, such as function generators and arbitrary waveform generators, and output amplifiers are characterized in a number of ways. Voltage histograms allow you to promptly characterize the sinusoidal output for voltage offset and distortion, especially zero crossing distortion or crossover distortion.

The voltage histogram of a sine waveform ideally looks like a saddle with the peaks occurring at the maxima and minima of the sine waveform. The saddle is centered at zero if the sine waveform has no DC voltage offset. Any anomalous peaks in the histogram between the maxima and minima reflect nonlinear distortion in the sine waveform. A peak at the center of the voltage histogram reflects zero-crossing distortion, which is a key concern for output amplifier performance.

The two following figures show voltage histograms taken using an NI 5122 high-speed digitizer. The histograms characterize the sine output of two function generators. The sine output in the first figure is relatively distortion-free and the

histogram centers at zero. By contrast, the histogram in the second figure is not centered at zero, indicating distortion and voltage drift in the sine output. Thus, while neither histogram shows zero-crossing distortion to be a primary problem, the noisy histogram in the second figure reveals considerable spectral purity degradation in the second function generator.



Voltage Histogram of 100 kHz 1 V_{pk-pk} Sine Output from Function Generator 1Voltage Histogram of 100 kHz 1 V_{pk-pk} Sine Output from Function Generator 2

Related Topics

[Creating Voltage Histograms](#)
 Creating Voltage Histograms

To create a [voltage histogram](#), you must use attributes and follow these general steps:

1. Set the size of the histogram with the [Voltage Histogram Size](#) property or the `NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_SIZE` attribute. This property defines the number of histogram bins. The larger the number of bins, the more resolution you have in the histogram. The default size is 256 bins.
2. Set the low limit of the voltage histogram with [Voltage Histogram Low Volts](#) property or the `NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_LOW_VOLTS` attribute. Set the high limit of the voltage histogram with [Voltage Histogram High Volts](#) property or the `NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_HIGH_VOLTS` attribute.

3. Call the [niScope Clear Waveform Measurement Stats VI](#) or the [niScope_ClearWaveformMeasurementStats](#) function to erase the histogram history.

Voltage Histogram Example (LabVIEW Only)

Refer to the `niScope_EX_Voltage_Histogram.vi` example for sample code you can use to create your own application.

Related Topics

[Voltage Histograms](#)

Coercions

NI-SCOPE allows you to configure the digitizer programmatically. However, the hardware may not support the exact value for some parameters that you specify. Instead of returning an error and forcing your program to be highly device specific, NI-SCOPE may coerce (or round) some input parameters to the next higher or lower value that the digitizer supports. Some of these coercions may result in incorrect behavior. In most cases, you can query the exact value of the parameter from NI-SCOPE.

Common coercions of NI-SCOPE parameters and attributes include [vertical](#), [horizontal](#), and [trigger](#) parameters.

Coercions of Horizontal Parameters

The horizontal timing parameters are all inter-related to comply with the IVI-Scope specification. These parameters all appear in the `Configure Horizontal Timing` function. The fundamental theory is that your time per record is a constant number during an acquisition. This is accomplished with the following parameter coercions:

Min Sample Rate

The `min sample rate` parameter is coerced up to the next available sample rate that the digitizer supports. The available sample rates change based on the acquisition type (such as normal and Flex Res). The actual sampling rate may be retrieved with

the `Sample Rate` function. This information is also returned by the `Fetch` function, with the `x` increment parameter, which is one divided by the actual sampling rate. If the sample rate parameter is set higher than the maximum, NI-SCOPE returns an error.

Enforce Realtime

If the `enforce realtime` parameter is set to `FALSE`, the digitizer enters RIS mode when the sampling rate exceeds the maximum real-time sampling rate of the device. In RIS mode, the sampling rate is coerced up to a multiple of the maximum real-time sampling rate.

Min Record Length

The `min record length` is coerced so that the time per record is constant. The formula is the following:

$$\text{actual rec length} = \text{min rec length} / \text{min sample rate} \times \text{actual sample rate}$$

You can find the actual record length by calling the `Actual Record Length` function. You can retrieve this value any time after you call `Configure Horizontal Timing`. Keep in mind that the amount of memory required on the digitizer is generally higher than the actual record length.

Reference Position

The `reference position` is rounded to the nearest sample. You can use the `relative initial x` parameter from a `Fetch` function to determine the actual reference position. If you fetch relative to the pretrigger, and the offset is zero, the `initial x` value is the first pretrigger point to the trigger. Therefore, the actual reference position is the following:

The following C code shows how NI-SCOPE accomplishes timing coercions:

```
// Convert minimum sample rate into its corresponding
// time-per-record value
```

```

timePerRecord = minRecordLength / minSampleRate;
if (minSampleRate <= maxRealTimeSampleRate)
{
    // For real-time, normal acquisitions, sample rate
    // is an integer divisor of max sample rate
    int divisor = (int) floor (maxRealTimeSampleRate / minS
ampleRate);
    actualSampleRate = maxRealTimeSampleRate / divisor;
}
else
{
    //In RIS, sample rate is a multiple of max sample rate
    int overSamplingFactor = (int) ceil (minSampleRate / ma
xRealTimeSampleRate);
    actualSampleRate = overSamplingFactor * maxRealTimeSamp
leRate;
}
actualRecordLength = (int) ((timePerRecord * actualSampleR
ate) + 0.5);

```

Coercions of Vertical Parameters

In the `Configure Vertical` function, vertical range is coerced to the next higher valid vertical range for the digitizer. You can determine the actual value with a `Binary Fetch` function.

 **Note** You can also use the `Vertical Range` attribute to determine the coerced value.

The gain scale factor is the volts/binary value. Therefore, if you use the `Fetch Binary 8` function, the actual vertical range is the gain scale factor times 256, with 256 being the number of binary values in an 8-bit number. If this vertical range is set higher than the maximum vertical range of the digitizer, NI-SCOPE returns an error.

In the `Configure Vertical` function, the vertical offset parameter is rounded to the nearest valid value, and it can also be obtained from the vertical offset output of the `Binary Fetch` function or from the `Vertical Offset` attribute.

Probe attenuation is applied exactly as specified. Therefore, if you select a 1.234 probe attenuation, your data is the voltage measured multiplied by 1.234. NI-SCOPE adjusts your vertical range based on the probe attenuation parameter, so the resulting voltage after the probe attenuation scaling is within the range specified. For example, if your probe attenuation is 10 and your vertical range is 10 V, the digitizer is set to measure a $1 V_{pk-pk}$ signal. The data returned with the Fetch function is $10 V_{pk-pk}$.

The max input frequency parameter in the `Configure Chan Characteristics` function enables or disables analog filters on the digitizer. Its value is coerced up to the next valid value. However, if this parameter is set to 0, the value is coerced to the default bandwidth for the digitizer. If this value is set higher than the maximum input frequency of the digitizer, NI-SCOPE returns an error.

The input impedance and vertical coupling parameters are not coerced. These parameters return an error if the value is not valid for the digitizer.

Coercions of Trigger Parameters

The trigger level, low level, high level, and hysteresis parameters found in the `Configure Trigger` functions are all coerced to the nearest valid value. Typically the hardware uses an 8-, 10-, or 12-bit DAC to set the trigger levels. The exact values cannot be queried for these parameters.

Using the Standard Functionality for error in Parameters

Many LabVIEW VIs contain `error in` parameters you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node. Standard error in behavior is as follows



Note Some nodes, such as error handling VIs, contain an `error in` parameter that does not provide standard error in functionality, but that contains an error in cluster that is standard.





`error in` describes error conditions that occur before this node runs. The default is `no error`. If an error occurred before this node runs, the node passes the `error in` value to `error out`. This node runs normally only if no error occurred before this node runs. If an error


occurs while this node runs, it runs normally and sets its own error status in error out.

Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

The error in cluster contains the following cluster elements:

 status is TRUE (X) if an error occurred before this node ran or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran. The default is FALSE.


 code is the error or warning code. The default is 0. If status is TRUE, code is an error code. If status is FALSE, code is 0 or a warning code.

 source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning. The default is an empty string.

Using the Standard Functionality for error out Parameters


Many LabVIEW VIs contain an error out parameter you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node.


Standard error out functionality is as follows:

 error out contains error information. If error in indicates that an error occurred before this VI ran, error out contains the same error information. Otherwise, it describes the error status that this VI produces. Right-click the error out front panel indicator and select Explain Error from the shortcut menu for more information about the error.

The error out contains the following cluster elements:

 status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

 code is the error or warning code. If status is TRUE, code is an error code. If status is FALSE, code is 0 or a warning code.

 source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning.

Array Measurements in NI-SCOPE

Array measurements are measurements on waveforms that result in an array of values such as an FFT amplitude spectrum. This section covers some key concepts needed to make array measurements.

The following is a categorized list of NI-SCOPE Array Measurements. The LabVIEW name is shown first, followed by the equivalent C/C++ name.

Histogram Measurements

LabVIEW	C/C++
Last Acq Histogram	NISCOPE_VAL_LAST_ACQ_HISTOGRAM
Multi Acq Time Histogram	NISCOPE_VAL_MULTI_ACQ_TIME_HISTOGRAM
Multi Acq Voltage Histogram	NISCOPE_VAL_MULTI_ACQ_VOLTAGE_HISTOGRAM

Window Measurements

LabVIEW	C/C++
Blackman Window	NISCOPE_VAL_BLACKMAN_WINDOW
Hanning Window	NISCOPE_VAL_HANNING_WINDOW
Flat Top Window	NISCOPE_VAL_FLAT_TOP_WINDOW
Triangle Window	NISCOPE_VAL_TRIANGLE_WINDOW
Hamming Window	NISCOPE_VAL_HAMMING_WINDOW

Filter Measurements

LabVIEW	C/C++
Bessel IIR Filter	NISCOPE_VAL_BESSEL_FILTER
FIR Windowed Filter	NISCOPE_VAL_WINDOWED_FIR_FILTER
Butterworth IIR Filter	NISCOPE_VAL_BUTTERWORTH_FILTER
Chebyshev IIR Filter	NISCOPE_VAL_CHEBYSHEV_FILTER

Two Channel Measurements

LabVIEW	C/C++
<u>Add Channels</u>	NISCOPE_VAL_ADD_CHANNELS
<u>Multiply Channels</u>	NISCOPE_VAL_MULTIPLY_CHANNELS
<u>Divide Channels</u>	NISCOPE_VAL_DIVIDE_CHANNELS
<u>Subtract Channels</u>	NISCOPE_VAL_SUBTRACT_CHANNELS

Math Measurements

LabVIEW	C/C++
<u>Derivative</u>	NISCOPE_VAL_DERIVATIVE
<u>Multi_Acq_Average</u>	NISCOPE_VAL_MULTI_ACQ_AVERAGE
<u>Array Integral</u>	NISCOPE_VAL_ARRAY_INTEGRAL
<u>Inverse</u>	NISCOPE_VAL_INVERSE
<u>Array Gain</u>	NISCOPE_VAL_ARRAY_GAIN
<u>Array Offset</u>	NISCOPE_VAL_ARRAY_OFFSET

FFT Measurements

LabVIEW	C/C++
<u>FFT Amp Spectrum (dB)</u>	NISCOPE_VAL_FFT_AMP_SPECTRUM_DB
<u>FFT Phase Spectrum</u>	NISCOPE_VAL_FFT_PHASE_SPECTRUM
<u>FFT Amp Spectrum (Volts RMS)</u>	NISCOPE_VAL_FFT_AMP_SPECTRUM_VOLTS_RMS

Resampling Measurements

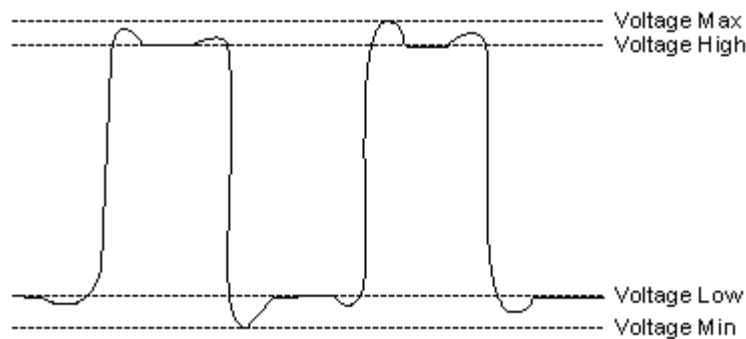
LabVIEW	C/C++
<u>Polynomial Interpolation</u>	NISCOPE_VAL_POLYNOMIAL_INTERPOLATION

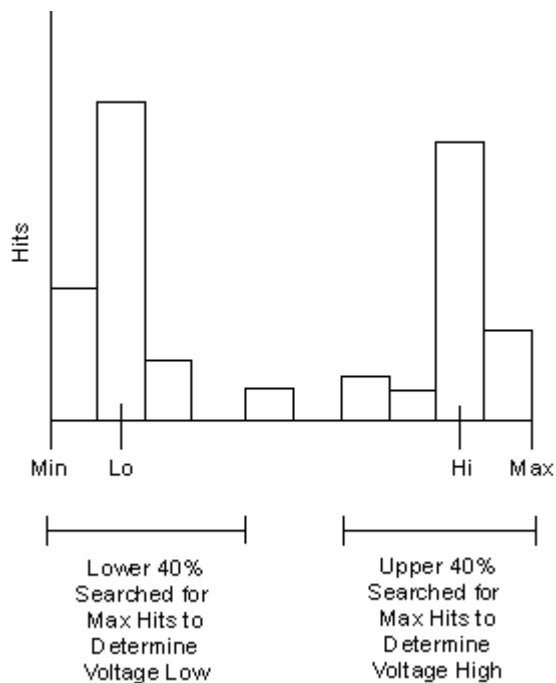
Last Acq Histogram

C/C++ Equivalent: NISCOPE_VAL_LAST_ACQ_HISTOGRAM

The initial x value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The x increment returned is the bin size. The last acquisition histogram method is used by the voltage low and voltage high measurements for computing the extrema of a waveform. This method is useful for ignoring overshoot or preshoot in a square waveform.

In the last acquisition histogram method, a voltage histogram is created from the most recent acquisition. The limits of the histogram are set by the minimum and maximum voltages of the acquisition. NI-SCOPE then counts how many samples fall into each bin of the histogram. The resolution is defined by the Last Acquisition Histogram Size attribute, which is 256 bins by default. You can fetch the histogram array using the Fetch Array Measurement function, with Last Acquisition Histogram constant specified as the array measurement function. The following illustrations show an example of a square wave and its resulting histogram. The voltage low and voltage high measurements correspond to the middle value of a bin with the maximum number of hits in the lower 40% or upper 40%, respectively.





Units

Count

X Units

Volts

Resulting Array Size

Measurement Last Acq Histogram Size Property

NI-SCOPE_ATTR_LAST_ACQ_HISTOGRAM_SIZE Attribute

Multi Acq Time Histogram

C/C++ Equivalent: `NISCOPE_VAL_MULTI_ACQ_TIME_HISTOGRAM`

The initial x value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The x increment returned is the bin size.

Units

Count

X Units

Seconds

Resulting Array Size

The value of the Measurement Time Histogram Size property or the NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_SIZE attribute during the first time histogram measurement after the histogram is cleared.

Multi Acq Voltage Histogram

C/C++ Equivalent: `NISCOPE_VAL_MULTI_ACQ_VOLTAGE_HISTOGRAM`

The initial x value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The x increment returned is the bin size.

Units

Count

X Units

Volts

Resulting Array Size

The value of the Measurement Voltage Histogram Size property or the NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_SIZE attribute during the first voltage histogram measurement after the histogram is cleared.

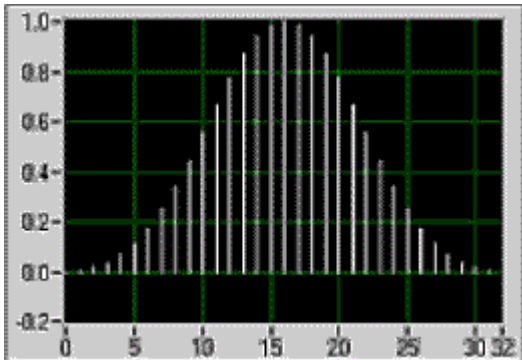
Blackman Window

C/C++ Equivalent: `NISCOPE_VAL_BLACKMAN_WINDOW`

A Blackman window is applied to the waveform using the following equation:

$$\mathbf{y}[\mathbf{i}] = \text{waveform}[\mathbf{i}] \times (0.42 - 0.50\cos(\mathbf{w}) + 0.08\cos(2\mathbf{w}))$$

where $\mathbf{w} = (2\pi)\mathbf{i}/\mathbf{n}$ and \mathbf{n} is the number of elements in the waveform.



Units

Volts

X Units

Seconds

Resulting Array Size

inputSize

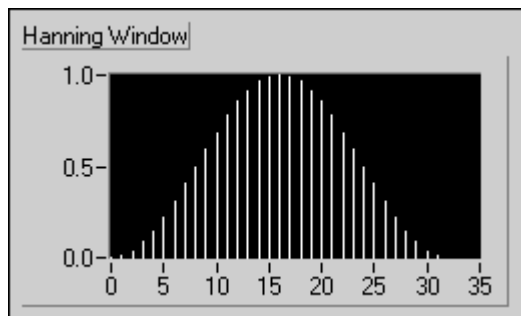
Hanning Window

C/C++ Equivalent: `NISCOPE_VAL_HANNING_WINDOW`

The Hanning window is useful for analyzing transients longer than the time duration of the window, and also for general-purpose applications. A Hanning window is applied to the waveform using the following equation:

$$\mathbf{y}[\mathbf{i}] = 0.5 \times \text{waveform}[\mathbf{i}] \times [1 - \cos(\mathbf{w})]$$

where $\mathbf{w} = (2\pi)\mathbf{i}/\mathbf{n}$ and \mathbf{n} = waveform size.



Note Hanning and Hamming windows are somewhat similar. However, in the time domain, the Hamming window does not get as close to zero near the edges as does the Hanning window.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize

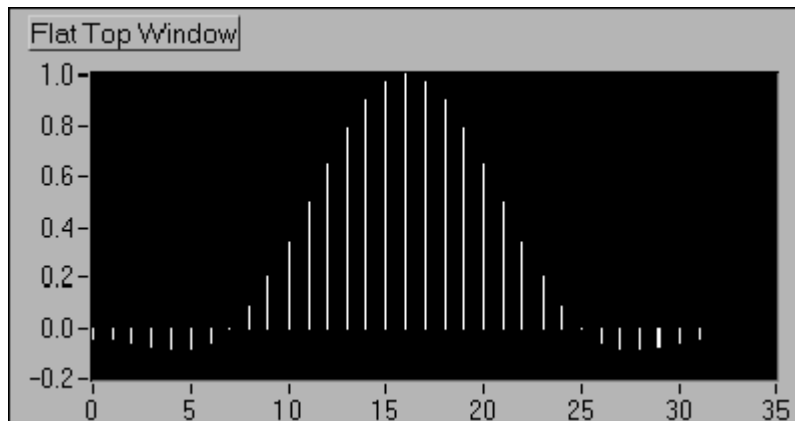
Flat Top Window

C/C++ Equivalent: `NISCOPE_VAL_FLAT_TOP_WINDOW`

The Flat Top window has the best amplitude accuracy of all the window functions. The increased amplitude accuracy (± 0.02 dB for signals exactly between integral cycles) is at the expense of frequency selectivity. The Flat Top window is most useful in accurately measuring the amplitude of single frequency components with little nearby spectral energy in the signal. A Flat Top window is applied to the waveform using the following equation:

$$\mathbf{y}[\mathbf{i}] = \text{waveform}[\mathbf{i}] \times (0.2810639 - 0.5208972\cos(\mathbf{w}) + 0.1980399\cos(2\mathbf{w}))$$

where $\mathbf{w} = (2\pi)\mathbf{i}/\mathbf{n}$ and \mathbf{n} is the waveform size.



Units

volts

X Units

Seconds

Resulting Array Size

inputSize

Triangle Window

C/C++ Equivalent: `NISCOPE_VAL_TRIANGLE_WINDOW`

A triangular window is applied to the waveform using the following equation:

$$\mathbf{y}[\mathbf{i}] = \text{waveform}[\mathbf{i}] \times (1 - |\mathbf{w}|)$$

where $\mathbf{w} = (2\mathbf{i} - \mathbf{n})/\mathbf{n}$ and \mathbf{n} is the number of elements in the waveform.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize

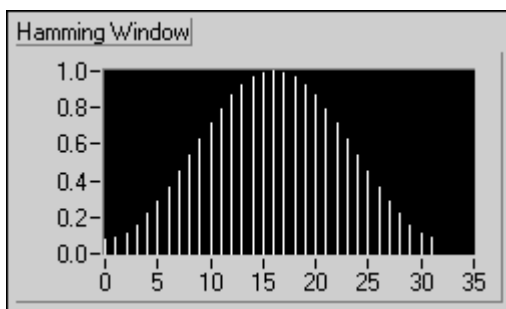
Hamming Window

C/C++ Equivalent: NISCOPE_VAL_HAMMING_WINDOW

A Hamming window is applied to the waveform using the following equation:

$$\mathbf{y}[i] = \mathbf{x}[i] [0.54 - 0.46\cos(\mathbf{w})]$$

where $\mathbf{w} = (2\pi)i/\mathbf{n}$ and \mathbf{n} = the number of elements in \mathbf{x} .



Note Hanning and Hamming windows are somewhat similar. However, in the time domain, the Hamming window does not get as close to zero near the edges as does the Hanning window.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize

Bessel IIR Filter

C/C++ Equivalent: NISCOPE_VAL_BESSEL_FILTER

The input waveform is filtered with a Bessel IIR algorithm. The type of the filter is specified by [Measurement Filter Type](#) as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The order of the filter is specified by the [Filter IIR Order](#) property or the `NISCOPE_ATTR_MEAS_FILTER_ORDER` attribute. Also, a number of points equal to the [Percent Waveform Transient](#) property or the `NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT` attribute divided by 100 times the input size are eliminated from the beginning of the filtered waveform so that the transient response of the filter does not affect further measurements.

Units

Volts

X Units

Seconds (nearly linear time delay)

Resulting Array Size

$\text{inputSize} \times (1 - \text{Percent Waveform Transient property} / 100)$

or

$\text{inputSize} \times (1 - \text{NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT attribute} / 100)$

Related Topics

[Bessel Filters](#)

FIR Windowed Filter

C/C++ Equivalent: `NISCOPE_VAL_WINDOWED_FIR_FILTER`

FIR filters are well suited for applications that require phase information, because the phase response is always linear. This linearity prevents distortion of time-domain filtered data. The transient response of an FIR filter is a fixed number of points, filter taps –1, which are eliminated in the filtered waveform returned from this VI.

Windowed FIR filters use the cutoff frequency as part of the design process, but the cutoff is specified before the windowing operation. For Windowed FIR filters of sufficient taps (more than 10 for a simple lowpass), the cutoff frequency has a magnitude near 0.5 (or 6 dB down). This magnitude varies slightly with the order and the window type.

The number of taps in the filter (that is, the number of coefficients in the filter) is given by the [Filter Taps](#) property or the `NISCOPE_ATTR_MEAS_FILTER_TAPS` attribute. This number must be odd for highpass and bandstop filters. The window for the coefficients is set with the [FIR Window](#) property or the `NISCOPE_ATTR_MEAS_FIR_FILTER_WINDOW` attribute.

Units

Volts

X Units

Seconds (linear time delay)

Resulting Array Size

$\text{inputSize} \times (\text{Filter Taps property} - 1)$

or

$\text{inputSize} \times (\text{NISCOPE_ATTR_MEAS_FILTER_TAPS attribute} - 1)$

Butterworth IIR Filter

C/C++ Equivalent: `NISCOPE_VAL_BUTTERWORTH_FILTER`

The input waveform is filtered with a Butterworth IIR algorithm. The type of the filter is specified by [Filter Type](#) as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The order of the filter is specified by the [Filter IIR Order](#) property or the `NISCOPE_ATTR_MEAS_FILTER_ORDER` attribute. Also, a number of points equal to [Percent Waveform Transient](#) property or the `NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT` attribute divided by 100 times the input size are eliminated from the beginning of the filtered waveform so the transient response of the filter does not affect further measurements.

Units

Volts

X Units

Seconds (nonlinear time delay)

Resulting Array Size

$\text{inputSize} \times (1 - \text{Percent Waveform Transient property} / 100)$

or

$\text{inputSize} \times (1 - \text{NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT attribute} / 100)$

Related Topics

[Butterworth Filters](#)

Chebyshev IIR Filter

C/C++ Equivalent: `NISCOPE_VAL_CHEBYSHEV_FILTER`

The input waveform is filtered with a Chebyshev IIR algorithm. The type of the filter is specified by the Filter Type property or the NISCOPE_ATTR_MEAS_FILTER_TYPE attribute as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The Chebyshev filter allows you to specify the passband ripple, the Filter Ripple or the NISCOPE_ATTR_MEAS_FILTER_RIPPLE, in decibels. The order of the filter is specified by the Filter Order property or the NISCOPE_ATTR_MEAS_FILTER_ORDER attribute. Also, a number of points equal to the Percent Waveform Transient property or the NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT attribute divided by 100 times the input size are eliminated from the beginning of the filtered waveform, so the transient response of the filter does not affect further measurements.

Units

Volts

X Units

Seconds (nonlinear time delay)

Resulting Array Size

$\text{inputSize} \times (1 - \text{Percent Waveform Transient property} / 100)$

or

$\text{inputSize} \times (1 - \text{NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM_PERCENT attribute} / 100)$

Related Topics

[Chebyshev Filters](#)

Add Channels

C/C++ Equivalent: `NISCOPE_VAL_ADD_CHANNELS`

Adds the waveforms from two channels, where one channel is specified by the channel parameter to the [niScope Fetch Measurement \(poly\) VI](#) (Measurement Scalar DBL instance) or the `niScope_FetchWaveformMeasurementArray` function, and the other channel is specified by the [Other Channel](#) property or the `NISCOPE_ATTR_MEAS_OTHER_CHANNEL` attribute. Any processing steps registered with the [niScope Add Waveform Processing VI](#) or the `niScope_AddWaveformProcessing` function are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize (on the channel specified by the measurement VI or function)

Multiply Channels

C/C++ Equivalent: `NISCOPE_VAL_MULTIPLY_CHANNELS`

Multiplies the waveforms from two channels, where one channel is specified by the channel parameter to the [niScope Fetch Measurement \(poly\) VI](#) (Measurement Scalar DBL instance) or the `niScope_FetchWaveformMeasurementArray` function, and the other channel is the [Other Channel](#) property or the `NISCOPE_ATTR_MEAS_OTHER_CHANNEL` attribute. Any processing steps registered with the [niScope Add Waveform Processing VI](#) or the `niScope_AddWaveformProcessing` function

are completed for the "other channel" before this measurement is taken. The two channels used in this measurement must be different.

Units

Volts²

X Units

Seconds

Resulting Array Size

inputSize (on the channel specified by the measurement VI)

Divide Channels

C/C++ Equivalent: `NISCOPE_VAL_DIVIDE_CHANNELS`

The channel specified by the channel parameter to [niScope Fetch Measurement \(poly\) VI](#) (Measurement Scalar DBL instance) or the or the `niScope_FetchWaveformMeasurementArray` function, is divided by the channel specified by the [Other Channel](#) property or the `NISCOPE_ATTR_MEAS_OTHER_CHANNEL` attribute. Any processing steps registered with the [niScope Add Waveform Processing VI](#) or the `niScope_AddWaveformProcessing` function are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

Units

None

X Units

Seconds

Resulting Array Size

inputSize (on the channel specified by the measurement VI or function)

Subtract Channels

C/C++ Equivalent: NISCOPE_VAL_SUBTRACT_CHANNELS

The channel specified by the [Other Channel](#) property or the `NISCOPE_ATTR_MEAS_OTHER_CHANNEL` attribute is subtracted from the channel specified by the channel parameter to [niScope Fetch Measurement \(poly\)](#) VI (Measurement Scalar DBL instance) or the `niScope_FetchWaveformMeasurementArray` function. Any processing steps registered with the [niScope Add Waveform Processing](#) VI or the `niScope_AddWaveformProcessing` function are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize (on the channel specified by the measurement VI or function)

Derivative

C/C++ Equivalent: NISCOPE_VAL_DERIVATIVE

The differences in the waveform are computed using the formula:

$y[i] = (\mathbf{waveform}[i+2] - \mathbf{waveform}[i]) / (2 \times \mathbf{dt})$, where **dt** is the time between two points.

Units

Volts/second

X Units

seconds

Resulting Array Size

inputSize - 2

Multi Acq Average

C/C++ Equivalent: NISCOPE_VAL_MULTI_ACQ_AVERAGE

The first time this measurement is called after it is cleared, an array the same size as the input is initialized to the input waveform, and the initial x and x increment values are set. Every subsequent call updates and returns the running average array without affecting the size of the array. The average array is cleared by calling [niScope Clear Waveform Measurement Stats VI](#) or the [niScope_ClearWaveformMeasurementStats](#) with the measurement function parameter set to Multi Acq Average.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize (during the first call to this measurement after clearing it)

Array Integral

C/C++ Equivalent: NISCOPE_VAL_ARRAY_INTEGRAL

The waveform is integrated using Simpson's rule, fitting a parabola to every 3 points with the equation:

$$y[i] = (\text{waveform}[i] + 4 \times \text{waveform}[i + 1] + \text{waveform}[i + 2]) \times dt/3$$

Units

Volts × Seconds

X Units

Seconds

Resulting Array Size

inputSize - 2

Inverse

C/C++ Equivalent: NISCOPE_VAL_INVERSE

The inverse is computed using the following formula:

$y[i] = 1.0/\text{waveform}[i]$. If waveform[i] is zero, the inverse is zero.

Units

1.0/volts

X Units

Seconds

Resulting Array Size

inputSize

Array Gain

C/C++ Equivalent: NISCOPE_VAL_ARRAY_GAIN

Every point in the waveform is multiplied by the scalar value Array Gain.

Units

Volts (if gain is unitless)

X Units

Seconds

Array Offset

C/C++ Equivalent: `NISCOPE_VAL_ARRAY_OFFSET`

The scalar value of measurement array offset is added to every point in the waveform.

Units

Volts (if offset is volts)

X Units

Seconds

FFT Amp Spectrum (dB)

C/C++ Equivalent: `NISCOPE_VAL_FFT_AMP_SPECTRUM_DB`

The amplitude spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two. The FFT is scaled to decibels:

$$\text{dB}[i] = 20.0 \times \log_{10} (\text{Amplitude}[i] / \text{Peak Amplitude})$$

Units

Decibels relative to peak

X Units

hertz

Resulting Array Size

Next higher power of two than the `inputSize` divided by two.

FFT Phase Spectrum

C/C++ Equivalent: `NISCOPE_VAL_FFT_PHASE_SPECTRUM`

The phase spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two. The phase spectrum is unwrapped.

Units

Radians

X Units

hertz

Resulting Array Size

Next higher power of two than `inputSize` divided by two.

FFT Amp Spectrum (Volts RMS)

C/C++ Equivalent: `NISCOPE_VAL_FFT_AMP_SPECTRUM_VOLTS_RMS`

The amplitude spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two.

Units

Volts RMS

X Units

hertz

Resulting Array Size

Next higher power of two than inputSize divided by two.

Polynomial Interpolation

C/C++ Equivalent: `NISCOPE_VAL_POLYNOMIAL_INTERPOLATION`

Polynomial interpolation allows oversampling or undersampling a waveform using any order polynomial set by the [Measurement Polynomial Interpolation Order](#) property or the `NISCOPE_ATTR_MEAS_POLYNOMIAL_INTERPOLATION_ORDER` attribute. For example, an order of 1 corresponds to linear interpolation. The new number of points is determined by the input size times the [Measurement Interpolation Sampling Factor](#) property or the `NISCOPE_ATTR_MEAS_INTERPOLATION_SAMPLING_FACTOR` attribute. The sampling factor can be any number greater than 0.

For every consecutive (order + 1) number of points, a polynomial is fit to the points, and new points are interpolated in a region of size **dx** in the middle of these order + 1 points, where **dx** is the original **x** spacing between two points in the waveform. This results in the interpolation shrinking the total **x** size of the waveform, since points cannot be accurately interpolated near the boundaries. In particular, the waveform loses $0.5 \times dx \times (\text{order} - 1)$ amount of **x** range at both the beginning and the end of the waveform.

Units

Volts

X Units

Seconds

Resulting Array Size

inputSize × Measurement Interpolation Sampling Factor property

or

inputSize × NISCOPE_ATTR_MEAS_INTERPOLATION_SAMPLING_FACTOR attribute

Scalar Measurements in NI-SCOPE

Scalar measurements are measurements on waveforms that produce a single value result. For example, you could use a scalar measurement to find the voltage amplitude or the median voltage of an acquisition.

The following is a categorized list of NI-SCOPE Scalar Measurements. The LabVIEW name is shown first, followed by the equivalent C/C++ name.

RMS Voltage Measurements

LabVIEW	C/C++
<u>AC Estimate</u>	NISCOPE_VAL_AC_ESTIMATE
<u>Voltage Cycle RMS</u>	NISCOPE_VAL_VOLTAGE_CYCLE_RMS
<u>FFT Amplitude</u>	NISCOPE_VAL_FFT_AMPLITUDE
<u>Voltage RMS</u>	NISCOPE_VAL_VOLTAGE_RMS

DC Voltage Measurements

LabVIEW	C/C++
<u>Voltage Average</u>	NISCOPE_VAL_VOLTAGE_AVERAGE
<u>Voltage Cycle Average</u>	NISCOPE_VAL_VOLTAGE_CYCLE_AVERAGE
<u>DC Estimate</u>	NISCOPE_VAL_DC_ESTIMATE

Area Measurements

LabVIEW	C/C++
<u>Area</u>	NISCOPE_VAL_AREA
<u>Integral</u>	NISCOPE_VAL_INTEGRAL
<u>Cycle Area</u>	NISCOPE_VAL_CYCLE_AREA

Voltage Extrema Measurements

LabVIEW	C/C++
<u>Voltage Min</u>	NISCOPE_VAL_VOLTAGE_MIN
<u>Voltage Low</u>	NISCOPE_VAL_VOLTAGE_LOW
<u>Voltage Max</u>	NISCOPE_VAL_VOLTAGE_MAX
<u>Voltage High</u>	NISCOPE_VAL_VOLTAGE_HIGH
<u>Voltage Peak to Peak</u>	NISCOPE_VAL_VOLTAGE_PEAK_TO_PEAK
<u>Voltage Amplitude</u>	NISCOPE_VAL_AMPLITUDE
<u>Voltage Base</u>	NISCOPE_VAL_VOLTAGE_BASE
<u>Voltage_Top</u>	NISCOPE_VAL_VOLTAGE_TOP
<u>Voltage Base to Top</u>	NISCOPE_VAL_VOLTAGE_BASE_TO_TOP
<u>Overshoot</u>	NISCOPE_VAL_OVERSHOOT
<u>Preshoot</u>	NISCOPE_VAL_PRESHOOT

Reference Levels

LabVIEW	C/C++
<u>Low Ref Volts</u>	NISCOPE_VAL_LOW_REF_VOLTS
<u>Mid Ref Volts</u>	NISCOPE_VAL_MID_REF_VOLTS
<u>High Ref Volts</u>	NISCOPE_VAL_HIGH_REF_VOLTS

Time Measurements

LabVIEW	C/C++
<u>Fall Time</u>	NISCOPE_VAL_FALL_TIME
<u>Positive Duty Cycle</u>	NISCOPE_VAL_DUTY_CYCLE_POS
<u>Falling Slew Rate</u>	NISCOPE_VAL_FALL_SLEW_RATE
<u>Rise Time</u>	NISCOPE_VAL_RISE_TIME
<u>Negative Width</u>	NISCOPE_VAL_WIDTH_NEG
<u>Rising Slew Rate</u>	NISCOPE_VAL_RISE_SLEW_RATE
<u>Positive Width</u>	NISCOPE_VAL_WIDTH_POS
<u>Negative Duty Cycle</u>	NISCOPE_VAL_DUTY_CYCLE_NEG

Two Channel Measurements

LabVIEW	C/C++
<u>Phase Delay</u>	NISCOPE_VAL_PHASE_DELAY
<u>Time Delay</u>	NISCOPE_VAL_TIME_DELAY

Period and Frequency Measurements

LabVIEW	C/C++
<u>Average Frequency</u>	NISCOPE_VAL_AVERAGE_FREQUENCY
<u>FFT Frequency</u>	NISCOPE_VAL_FFT_FREQUENCY
<u>Average Period</u>	NISCOPE_VAL_AVERAGE_PERIOD
<u>Frequency</u>	NISCOPE_VAL_FREQUENCY
<u>Period</u>	NISCOPE_VAL_PERIOD

Time Histogram Measurements

LabVIEW	C/C++
<u>Time Hist Hits</u>	NISCOPE_VAL_TIME_HISTOGRAM_HITS
<u>Time Hist Mean + 3 Stdev</u>	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_3_STDEV
<u>Time Hist Max</u>	NISCOPE_VAL_TIME_HISTOGRAM_MAX
<u>Time Hist Median</u>	NISCOPE_VAL_TIME_HISTOGRAM_MEDIAN
<u>Time Hist Mean</u>	NISCOPE_VAL_TIME_HISTOGRAM_MEAN
<u>Time Hist Min</u>	NISCOPE_VAL_TIME_HISTOGRAM_MIN
<u>Time Hist Peak to Peak</u>	NISCOPE_VAL_TIME_HISTOGRAM_PEAK_TO_PEAK
<u>Time Hist Mode</u>	NISCOPE_VAL_TIME_HISTOGRAM_MODE
<u>Time Hist Mean + Stdev</u>	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_STDEV
<u>Time Hist New Hits</u>	NISCOPE_VAL_TIME_HISTOGRAM_NEW_HITS
<u>Time Hist Mean + 2 Stdev</u>	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_2_STDEV
<u>Time Hist Stdev</u>	NISCOPE_VAL_TIME_HISTOGRAM_STDEV

Voltage Histogram Measurements

LabVIEW	C/C++
<u>Volt Hist Hits</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_HITS
<u>Volt Hist Median</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEDIAN
<u>Volt Hist Max</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MAX
<u>Volt Hist Min</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MIN
<u>Volt Hist Mean</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN
<u>Volt Hist Mode</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MODE
<u>Volt Hist Mean + Stdev</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_STDEV
<u>Volt Hist New Hits</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_NEW_HITS
<u>Volt Hist Mean + 2 Stdev</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_2_STDEV
<u>Volt Hist Peak to Peak</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_PEAK_TO_PEAK
<u>Volt Hist Mean + 3 Stdev</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_3_STDEV
<u>Volt Hist Stdev</u>	NISCOPE_VAL_VOLTAGE_HISTOGRAM_STDEV

AC Estimate

C/C++ Equivalent: NISCOPE_VAL_AC_ESTIMATE

The DC estimate is subtracted from the waveform, and a Hanning window is applied to give a processed waveform. The RMS voltage is calculated with the following equation:

$$\text{sqrt} \left(\left[\sum \text{processed waveform}[i]^2 \right] / \left[\text{numPoints} \times \text{enbw} \times \text{cg}^2 \right] \right),$$

where the equivalent noise bandwidth (**enbw**) for the Hanning window is 1.5, and the coherent gain (**cg**) is 0.5.

This algorithm minimizes the effect of an uneven number of waveform cycles in the measurement, which could arbitrarily increase or decrease the RMS value.

Units

Volts RMS

Voltage Cycle RMS

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_CYCLE_RMS

Voltage Cycle RMS = $\sqrt{(\sum \text{waveform}[i]^2) / \text{pointsPerPeriod}}$

The number of points in a period is calculated using the equation:

$\text{pointsPerPeriod} = \text{period} / dt$

where **period** is the measured period of the signal and **dt** is the time between two points. The result will then be converted to an integer from a floating point value.

Units

Volts RMS

Voltage Cycle Average

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_CYCLE_AVERAGE

Voltage Cycle Average = $(\sum \text{waveform}[i]) / \text{pointsPerPeriod}$

The number of points in a period in volts is calculated using the equation:

$\text{pointsPerPeriod} = \text{period} / dt$

where **period** is the measured period of the signal and **dt** is the time between two points. The result will then be converted to an integer from a floating point value.

FFT Amplitude

C/C++ Equivalent: NISCOPE_VAL_FFT_AMPLITUDE

The FFT amplitude spectrum is calculated using a split-radix real FFT, and the maximum amplitude is returned. If the input waveform size is not a power of two, the waveform is zero-padded to the next higher power of two. For best results, verify that your actual record length is a power of 2.

Units

Volts RMS

Voltage RMS

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_RMS

Determined by the following equation:

$$\mathbf{Voltage\ RMS} = \sqrt{[(\sum \mathbf{waveform}[i]^2) / \mathbf{numPoints}]}$$

Units

Volts RMS

Voltage Average

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_AVERAGE

Determined by the following equation:

$$\mathbf{Voltage\ Average} = \sum \mathbf{waveform}[i] / \mathbf{numPoints}$$

Units

Volts

DC Estimate

C/C++ Equivalent: NISCOPE_VAL_DC_ESTIMATE

A Hanning window is applied to give a "processed waveform" and the voltage average is calculated with the following equation:

voltage average = $(\sum \text{processed waveform}[i]) / (\text{cg} \times \text{numPoints})$,

where the coherent gain (**cg**) of the Hanning window is 0.5 – the DC gain of the window.

The algorithm minimizes the effect of an uneven number of waveform cycles. For example, performing a simple voltage average on 5.5 cycles of a sine waveform gives a slightly incorrect DC estimate if the extra half cycle is not evenly divided between the positive and negative portions of the sine wave.

Units

Volts

Area

C/C++ Equivalent: NISCOPE_VAL_AREA

area = voltage average × **numPoints** × **delta time** between two points.

Units

Volts × Seconds

Integral

C/C++ Equivalent: NISCOPE_VAL_INTEGRAL

Numerical integration is done using Simpson's rule.

Units

Volts × seconds

Cycle Area

C/C++ Equivalent: NISCOPE_VAL_CYCLE

Cycle area = **voltage cycle average** × **pointsPerPeriod** × **dt**

where **pointsPerPeriod** is defined in the voltage cycle average description.

Units

Volts × Seconds

Voltage Min

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_MIN

The waveform is searched for the minimum point in volts.

Voltage Low

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_LOW

The last acquisition histogram method is used where the voltage low result is the voltage of the histogram bin with the maximum number of hits below 40% of the waveform's voltage peak-to-peak value. This calculation is useful for ignoring the overshoot and preshoot on square waves.

Units

Volts

Voltage Max

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_MAX

The waveform is searched for its maximum point.

Units

Volts

Voltage High

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HIGH

The last-acquisition histogram method is used, where the voltage high result is the voltage of the histogram bin with the maximum number of hits above 60% of the

waveform's voltage peak-to-peak value. This calculation is useful for ignoring the overshoot and preshoot on square waves.

Units

Volts

Voltage Amplitude

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_AMPLITUDE

The voltage high minus the voltage low.

Units

Volts

Voltage Peak to Peak

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_PEAK_TO_PEAK.

The maximum voltage minus the minimum voltage in volts.

Voltage Base

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_BASE

If the histogram bin corresponding to voltage low has over five percent of the total hits, the voltage low result is returned. Otherwise, the voltage minimum calculation is returned. This allows using the voltage base to get a reasonable answer for either a square wave (ignoring the overshoot and preshoot) or a triangle wave (where a histogram fails).

Units

Volts

Voltage Top

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_TOP

If the histogram bin corresponding to voltage high has over five percent of the total hits, the voltage high result is returned. Otherwise, the voltage maximum calculation is returned. This allows using the voltage top to get a reasonable answer for either a square wave (ignoring the overshoot and preshoot) or a triangle wave (where a histogram fails).

Units

Volts

Voltage Base to Top

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_BASE_TO_TOP

Voltage top minus voltage base.

Units

Volts

Overshoot

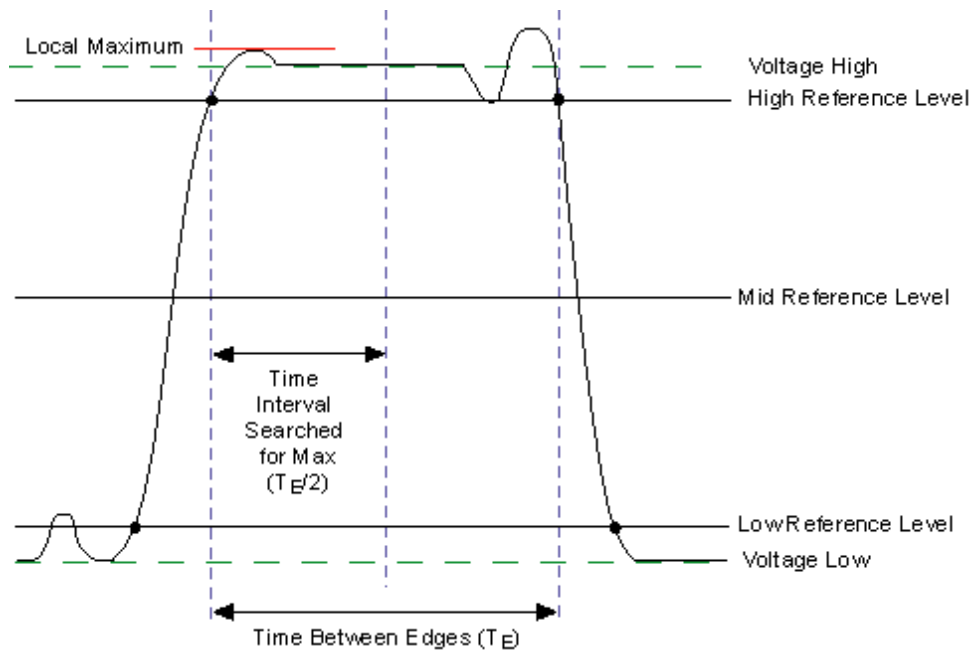
C/C++ Equivalent: NISCOPE_VAL_OVERSHOOT

The measurement is taken on the first edge of the waveform. If two edges exist, the algorithm finds the time interval from the first edge until one half the time to the second edge. The local maxima and minima are found in this interval. If only one edge is present in the waveform, the local maximum and minimum is found between the first edge and the end of the waveform.

If the first edge is positive sloped, **overshoot** = $100 \times (\text{local maximum} - \text{voltage high}) / \text{voltage amplitude}$.

If the first edge is negative sloped, **overshoot** = $100 \times (\text{voltage low} - \text{local minimum}) / \text{voltage amplitude}$.

The following figure shows an example of an overshoot measurement.



Units

Percentage

Preshoot

C/C++ Equivalent: `NISCOPE_VAL_PRESHOOT`

The measurement is taken on the second edge of the waveform if two edges exist. The algorithm finds the time interval from the middle time between the two edges until the second edge. The local maxima and minima are found in this interval. If only one edge is present in the waveform, the local maximum and minimum are found from the start of the waveform to the first edge.

If the edge is negative sloped, **preshoot** = $100 \times (\text{local maximum} - \text{voltage high}) / \text{voltage amplitude}$.

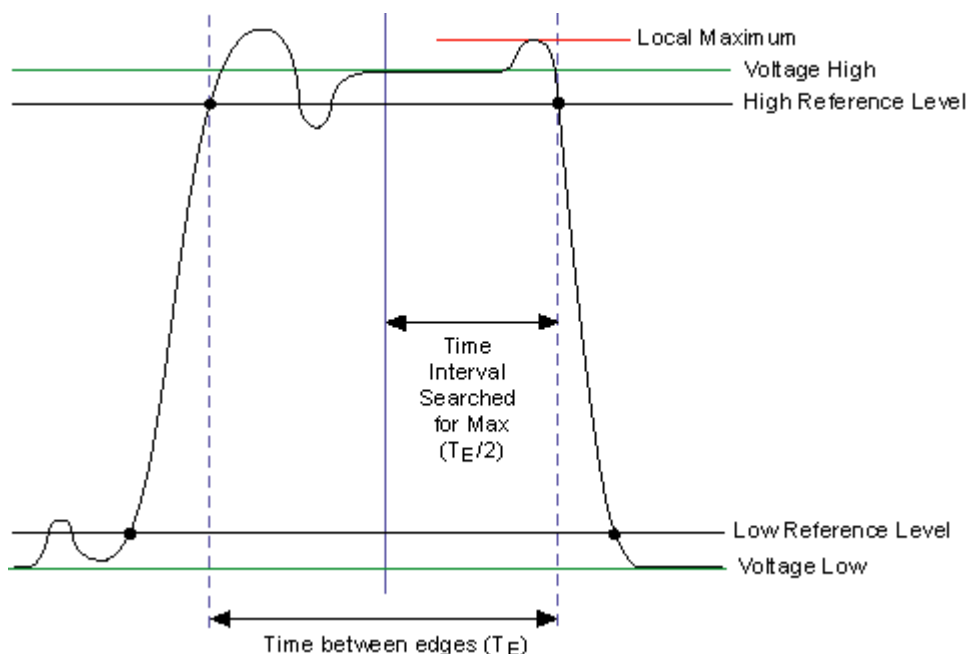
If the edge is positive sloped, **preshoot** = $100 \times (\text{voltage low} - \text{local minimum}) / \text{voltage amplitude}$.

Units

Percentage

Details

The following figure shows an example of a preshoot measurement.



Low Ref Volts

C/C++ Equivalent: `NISCOPE_VAL_LOW_REF_VOLTS`

The voltage corresponding to the low reference level. If the [Reference Level Units](#) property or the [NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS](#) attribute is set to Voltage, the value of the [Channel Based Low Ref Level](#) property or the [NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL](#) attribute is returned.

If the Reference Level Units property or the `NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS` attribute is set to Percentage, the voltage is calculated with the method specified by the [Percentage Units Method](#) property or the [NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD](#) attribute.

Units

Volts

Mid Ref Volts

C/C++ Equivalent: `NISCOPE_VAL_MID_REF_VOLTS`

The voltage corresponding to the mid reference level. If the [Reference Level Units](#) property or the [NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS](#) attribute is set to Voltage, the value of the [Channel Based Mid Ref](#) property or the [NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL](#) attribute is returned.

If the Reference Level Units property or the `NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS` attribute is set to Percentage, the voltage is calculated with the method specified by the [Percentage Units Method](#) property or the [NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD](#) attribute.

Units

Volts

High Ref Volts

C/C++ Equivalent: `NISCOPE_VAL_HIGH_REF_VOLTS`

The voltage corresponding to the high reference level. If the [Reference Level Units](#) property or the [NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS](#) attribute is set to Voltage, the value of the [Channel Based High Ref](#) property or the [NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL](#) attribute is returned.

If the Reference Level Units property or the `NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS` attribute is set to Percentage, the voltage is calculated with the method specified by the [Percentage Units Method](#) property or the [NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD](#) attribute.

Units

Volts

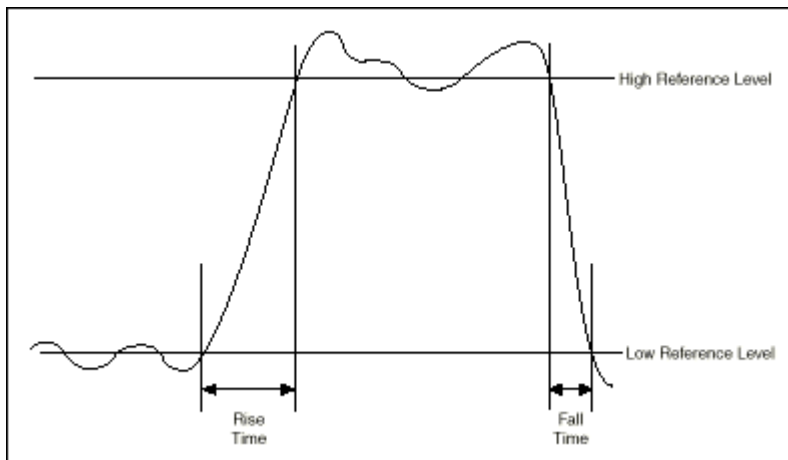
Fall Time

C/C++ Equivalent: `NISCOPE_VAL_FALL_TIME`

The time span in seconds from when the waveform crosses the high reference level until it crosses the low reference level. The measurement starts at the left edge of the waveform and finds all high reference level crossings until a low reference level crossing. The final high reference level crossing is used in the calculation.

The reference levels are specified by [Channel Based Low Ref](#) and [Channel Based High Ref](#), and their default values are 10% and 90%.

The following figure shows an example of fall time.



Units

Seconds

Positive Duty Cycle

C/C++ Equivalent: `NISCOPE_VAL_DUTY_CYCLE_POS`

The [positive width](#) divided by the [period](#) times 100.

Units

Percentage

Falling Slew Rate

C/C++ Equivalent: `NISCOPE_VAL_FALL_SLEW_RATE`

The low reference voltage minus the high reference voltage is divided by the fall-time calculation. The result is always negative.

Units

Volts/second

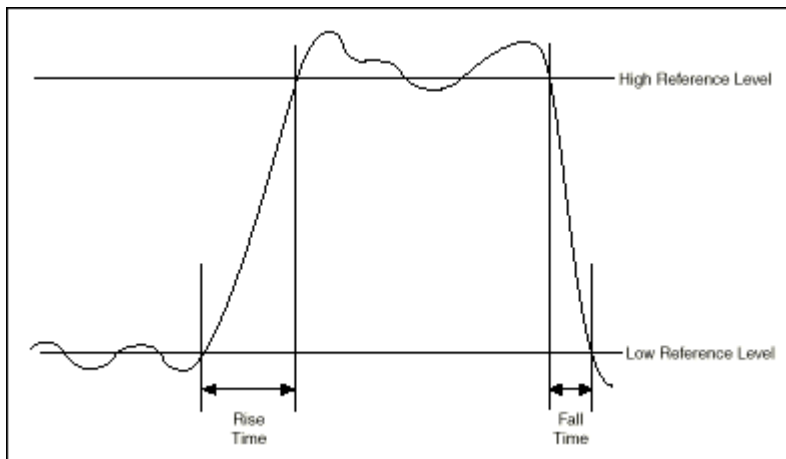
Rise Time

C/C++ Equivalent: `NISCOPE_VAL_RISE_TIME`

The time span in seconds from when the waveform crosses the low reference level until it crosses the high reference level. The measurement starts at the left edge of the waveform and finds all low reference level crossings until a high reference level crossing. The final low reference level crossing is used in the calculation.

The reference levels are specified by Channel Based Low Ref and Channel Based High Ref, and their default values are 10% and 90%.

The following figure shows an example of rise time.



Units

Seconds

Negative Width

C/C++ Equivalent: NISCOPE_VAL_WIDTH_NEG

The time difference between the first two mid reference level crossings, where the slopes are negative and positive, respectively. A digital hysteresis is used when finding the crosspoints.

Units

Seconds

Rising Slew Rate

C/C++ Equivalent: NISCOPE_VAL_RISE_SLEW_RATE

The high reference voltage minus the low reference voltage is divided by the rise-time calculation.

Units

Volts/second

Positive Width

C/C++ Equivalent: NISCOPE_VAL_WIDTH_POS

The time difference in seconds between the first two mid reference level crossings, where the slopes are positive and negative respectively. A digital hysteresis is used when finding the crosspoints.

Units

Seconds

Negative Duty Cycle

C/C++ Equivalent: NISCOPE_VAL_DUTY_CYCLE_NEG

The negative width divided by the period times 100.

Units

Percentage

Phase Delay

C/C++ Equivalent: `NISCOPE_VAL_PHASE_DELAY`

The [time delay](#) divided by the period (of the waveform on the channel specified by the measurement VI or function) times 360 degrees.

Units

Degrees

Time Delay

C/C++ Equivalent: `NISCOPE_VAL_TIME_DELAY`

The time between a rising or falling edge on one channel and the subsequent edge, by default of the same type, on a second channel.

The algorithm operates as follows:

1. Finds the first time that the waveform from the channel specified by the channels parameter crosses its mid-reference level.
2. Finds the first two times that the waveform from the channel the [Other Channel](#) property or the [NISCOPE_ATTR_MEAS_OTHER_CHANNEL](#) attribute crosses its mid-reference level.
3. By default, measures the time delay across channels as rising edge to rising edge or falling edge to falling edge, depending on which edge type the algorithm first detects on the first channel.



Tip You can invert the data from either or both channels by adding a processing step and using the array measurement gain with a value of -1. With this method, you can also measure from rising edge to falling edge or from falling edge to rising edge by inverting a signal on only one of the two channels.

The mid-reference level is stored on a per-channel basis, and mid-reference levels do not need to be the same. All reference levels use a digital hysteresis.

Units

Seconds

Average Frequency

C/C++ Equivalent: `NISCOPE_VAL_AVERAGE_FREQUENCY`

1.0 divided by the [average period](#).

Units

hertz

FFT Frequency

C/C++ Equivalent: `NISCOPE_VAL_FFT_FREQUENCY`

The FFT amplitude spectrum is calculated using a split-radix real FFT, and the frequency corresponding to the maximum amplitude is returned. If the input waveform size is not a power of two, the waveform is zero padded to the next higher power of two. The frequency resolution is sampling rate / number of points.

The DC bin of the FFT is ignored when searching for the maximum amplitude, so the FFT frequency should ignore any DC offsets. However, the zero padding used in the FFT measurement can introduce other low frequency components if the waveform has a large DC offset. To avoid problems, make sure the [Horizontal Actual Record Length](#) property or the `NISCOPE_ATTR_HORZ_RECORD_LENGTH` attribute is a power of 2, so no zero padding occurs. This property/attribute can be fetched using the [niScope Actual Record Length VI](#) or the [niScope_ActualRecordLength](#) function. Alternatively, configuring the digitizer for AC coupling solves the problem.

Units

hertz

Average Period

C/C++ Equivalent: NISCOPE_VAL_AVERAGE_PERIOD

Up to 256 mid reference level crossings are found on the waveform, using a digital hysteresis. The time difference between the last crossing and the first crossing is divided by the number of periods found in the waveform. The last crossing is defined as the last crossing in the waveform with the same slope as the first crossing, so an integer number of periods exist in the waveform.

Units

Seconds

Frequency

C/C++ Equivalent: NISCOPE_VAL_FREQUENCY

1.0 divided by the period, in hertz.

Period

C/C++ Equivalent: NISCOPE_VAL_PERIOD

Finds the time in seconds between the first and third mid reference level crosspoints. A hysteresis window is applied when finding crosspoints. The mid reference level is 50% by default and is set with [Channel Based Mid Ref Level](#).

Units

Seconds

Time Hist Hits

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_HITS

Number of points in the histogram.

Time Hist Max

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_MAX

The highest bin value with at least one hit.

Units

Seconds

Time Hist Mean

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_MEAN

Histogram Mean = $[\sum (\text{bin hits} \times \text{bin value})] / \text{time hist hits}$.

The bin value is the center time value of the histogram bin.

Units

Seconds

Time Hist Mean + Stdev

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_STDEV

The percentage of hits in the histogram between mean minus the standard deviation and mean plus the standard deviation. The percentage is returned in the range 0–100.

Units

Percentage

Time Hist Mean + 2 Stdev

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_2_STDEV

The percentage of hits in the histogram between the mean minus two times the standard deviation and the mean plus two times the standard deviation. The percentage is returned in the range 0—100.

Units

Percentage

Time Hist Mean + 3 Stdev

C/C++ Equivalent: `NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_3_STDEV`

The percentage of hits in the histogram between the mean minus three times the standard deviation and the mean plus three times the standard deviation. The percentage is returned in the range 0—100.

Units

Percentage

Time Hist Median

C/C++ Equivalent: `NISCOPE_VAL_TIME_HISTOGRAM_MEDIAN`

The bin value where half the histogram hits are above it and half the histogram hits are below.

Units

Seconds

Time Hist Min

C/C++ Equivalent: `NISCOPE_VAL_TIME_HISTOGRAM_MIN`

The lowest bin value with at least one hit.

Units

Seconds

Time Hist Mode

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_MODE

The bin value with the most hits. If there is a tie, the lower voltage or time value is returned.

Units

Seconds

Time Hist New Hits

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_NEW_HITS

Number of points added to the histogram by the most recent acquisition.

Time Hist Peak to Peak

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_PEAK_TO_PEAK

Histogram maximum minus the histogram minimum.

Units

Seconds

Time Hist Stdev

C/C++ Equivalent: NISCOPE_VAL_TIME_HISTOGRAM_STDEV

$$\text{Histogram Stdev} = \sqrt{\sum[\text{bin hits} \times (\text{bin value} - \text{histogram mean})^2] / (\text{total hits} - 1)}$$

Units

Seconds

Volt Hist Hits

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_HITS

Number of points in the histogram.

Volt Hist Max

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MAX

The highest bin value with at least one hit.

Units

Volts

Volt Hist Mean

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN

Histogram Mean = [(**bin hits** × **bin value**)] / volt hist hits.

The bin value is the center voltage value of the histogram bin.

Units

Volts

Volt Hist Mean + Stdev

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_STDEV

The percentage of hits in the histogram between mean minus the standard deviation and mean plus the standard deviation. The percentage is returned in the range 0–100.

Units

Percentage

Volt Hist Mean + 2 Stdev

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_2_STD
EV

The percentage of hits in the histogram between the mean minus two times the standard deviation and the mean plus two times the standard deviation. The percentage is returned in the range 0–100.

Units

Percentage

Volt Hist Mean + 3 Stdev

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_3_STD
EV

The percentage of hits in the histogram between the mean minus three times the standard deviation and the mean plus three times the standard deviation. The percentage is returned in the range 0–100.

Units

Percentage

Volt Hist Median

C/C++ Equivalent: NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEDIAN

The bin value where half the histogram hits are above it and half the histogram hits are below.

Units

Volts

Volt Hist Min

C/C++ Equivalent: `NISCOPE_VAL_VOLTAGE_HISTOGRAM_MIN`

The lowest bin value with at least one hit.

Units

Volts

Volt Hist Mode

C/C++ Equivalent: `NISCOPE_VAL_VOLTAGE_HISTOGRAM_MODE`

The bin value with the most hits. If there is a tie, the lower voltage or time value is returned.

Units

Volts

Volt Hist New Hits

C/C++ Equivalent: `NISCOPE_VAL_VOLTAGE_HISTOGRAM_NEW_HITS`

Number of points added to the histogram by the most recent acquisition.

Volt Hist Peak to Peak

C/C++ Equivalent: `NISCOPE_VAL_VOLTAGE_HISTOGRAM_PEAK_TO_PEAK`

Histogram maximum minus the histogram minimum.

Units

Volts

Volt Hist Stdev

C/C++ Equivalent: `NISCOPE_VAL_VOLTAGE_HISTOGRAM_STDEV`

$$\text{Histogram Stdev} = \sqrt{\sum[\text{bin hits} \times (\text{bin value} - \text{histogram mean})^2] / (\text{total hits} - 1)}$$

Units

Volts

NI-SCOPE Instrument Driver FPGA Extensions

NI-SCOPE instrument driver FPGA extensions for reconfigurable oscilloscopes enable you to use precompiled FPGA bitfiles to customize the behavior of a reconfigurable oscilloscope FPGA while maintaining the functionality of the NI-SCOPE instrument driver.

Before you begin using NI-SCOPE instrument driver FPGA extensions, you must install NI-SCOPE and LabVIEW. You must also download a precompiled bitfile to use NI-SCOPE instrument driver FPGA extensions. Visit ni.com/info and enter `ex6h8h` as the Info Code, then select your hardware, and browse the download page to download a NI-SCOPE FPGA extensions precompiled FPGA bitfile.



Note Certain NI-SCOPE instrument driver FPGA extensions might require you to install the instrument design libraries. Refer to the installation requirements of individual NI-SCOPE instrument driver FPGA extensions located on the download page for more information.

The precompiled bitfiles are included in examples that illustrate how to use the precompiled bitfiles. These examples contain three parts: a precompiled bitfile featuring application-specific IP, a host-based example that uses NI-SCOPE, and IP dependencies, such as source code. These examples are distributed through the VI Package Manager, which installs with LabVIEW.

When you install NI-SCOPE, model-specific folders for the supported devices are created in the following location:

- **Windows 10/8.1/7:** `Users\Public\Documents\National Instruments\FPGA Extensions Bitfiles`



Note The instrument driver searches for precompiled bitfiles in these model-specific folders. For the instrument driver to recognize your precompiled bitfile, ensure that the precompiled bitfile is located in the model-specific folder that corresponds to your device.

Related Topics

[Using NI-SCOPE Instrument Driver FPGA Extensions](#)

Visit ni.com/info and enter `exkt9h` as the Info Code to learn more about instrument driver FPGA extensions.

Using NI-SCOPE Instrument Driver FPGA Extensions

Before you begin using NI-SCOPE instrument driver FPGA extensions, you must install NI-SCOPE and LabVIEW. When you install NI-SCOPE, model-specific folders for the supported devices are created in the following location:

- **Windows 10/8.1/7:** `Users\Public\Documents\National Instruments\FPGA Extensions Bitfiles`



Note The instrument driver searches for precompiled bitfiles in these model-specific folders. In order for the instrument driver to recognize your precompiled bitfile, ensure that the precompiled bitfile is located in the model-specific folder that corresponds to your device.

To use NI-SCOPE instrument driver FPGA extensions, you must specify the precompiled bitfile you want to use when you initialize the instrument driver session. If you want to access additional FPGA functionality provided by the precompiled bitfile, you must also obtain an FPGA reference to use with the LabVIEW FPGA host interface.

Complete the following steps to open an instrument driver session that uses a precompiled bitfile:

1. Call the `niScope Initialize with Options VI`.
2. Wire the `Driver Setup` string to the `options` string input.
3. Include the `Bitfile` tag in the `Driver Setup` string, and set the value of the `Bitfile` tag to the name of the bitfile you want to load, as shown in the following example string:

```
DriverSetup=Bitfile:filename.lvbitx
```

where `filename.lvbitx` is the file name of the bitfile you want to load.

Complete the following steps to obtain an FPGA reference to use with the LabVIEW FPGA host interface:

1. Call the niScope FPGA Bitfile Path property. The FPGA Bitfile Path property returns the absolute path to the bitfile as a string.
2. Wire the output of the FPGA Bitfile Path property to the String input of the String to Path function.
3. Wire the Path output of the String to Path function to the Bitfile Path input of the Open Dynamic Bitfile Reference function.

Related Topics

Visit ni.com/info and enter `ex6h8h` as the Info Code to download precompiled NI-SCOPE instrument driver FPGA extensions bitfiles.

Visit ni.com/info and enter `exkt9h` as the Info Code to learn more about instrument driver FPGA extensions.

NI-SCOPE LabVIEW Reference

Expand this topic to view the VIs and properties included with NI-SCOPE that you can use to configure and operate your NI high-speed digitizer.

NI-SCOPE VIs

July 2019

Use the NI-SCOPE VIs to develop applications for your NI high-speed digitizer.

The following tables show an overview of the VIs and subpalettes on the main NI-SCOPE palette.

Palette Object	Description
niScope Initialize	Performs the following initialization actions: <ul style="list-style-type: none"> -Creates a new IVI instrument driver session. -Opens a session to the device(s) that you specify in the resource name parameter.

	<ul style="list-style-type: none"> -Queries each instrument ID and verifies that it is valid for this instrument driver. -Resets the instrument(s) to a known state if the reset device parameter is set to TRUE; refer to niScope Reset for the default state of each digitizer. -Sends initialization commands to set the instrument(s) to the state necessary for the operation of the instrument driver. -Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver VI calls.
<u>niScope Configure Vertical</u>	Configures the most commonly configured properties of the digitizer vertical subsystem, such as the range, offset, coupling, probe attenuation, and the channel name.
<u>niScope Configure Horizontal Timing</u>	Configures the common properties of the horizontal subsystem for a single record or multirecord acquisition, such as minimum sample rate and acquisition size.
<u>niScope Configure Trigger (poly)</u>	Configures the digitizer for different types of triggering.
<u>niScope Read (poly)</u>	Initiates an acquisition, waits for it to complete, and acquires data.
<u>niScope Close</u>	<p>Performs the following actions:</p> <ul style="list-style-type: none"> -Closes the instrument I/O session. -Destroys the IVI session and all of its properties. -Deallocates any memory resources used by the IVI session. <p>Call this VI when you are finished using an instrument driver session.</p>
<u>niScope Initialize With Options</u>	Performs the following initialization actions:

	<p>-Creates a new IVI instrument driver session and optionally sets the initial state of the following session properties: Range Check, Cache, Simulate, Record Value Coercions.</p> <p>-Opens a session to the device(s) that you specify in the resource name parameter.</p> <p>-Queries each instrument ID and verifies that it is valid for this instrument driver.</p> <p>-Resets the instrument(s) to a known state if the reset device parameter is set to TRUE; refer to niScope Reset for the default state of each digitizer.</p> <p>-Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver VI calls.</p>
--	---

Subpalette	Description
Configuration	Use the NI-SCOPE Configuration VIs to set up and transfer the parameters of your acquisition, or use Auto Setup to automatically configure device settings.
Timing	Use the NI-SCOPE Timing VIs to configure common timing properties for your digitizer.
Triggering	Use the NI-SCOPE Triggering VIs to configure the trigger properties.
Acquisition	Use the NI-SCOPE Acquisition VIs to control your data acquisition and to retrieve data from your device.
Calibration	Use the NI-SCOPE Calibration VIs to self-calibrate your digitizer and to access lower-level external calibration functions.
Measurements	Use the NI-SCOPE Measurements VIs to perform waveform measurements, including scalar and array measurements.

Utility Functions

Use the NI-SCOPE Utility VIs to perform various tasks such as resetting the digitizer and returning the revision numbers of the driver and the instrument firmware.

© 2001–2019 National Instruments Corporation. All Rights Reserved.

niScope Initialize

Performs the following initialization actions:


- Creates a new IVI instrument driver session.
- Opens a session to the device(s) that you specify in the resource name parameter.
- Queries each instrument ID and verifies that it is valid for this instrument driver.
- Resets the instrument(s) to a known state if the reset device parameter is set to TRUE; refer to niScope Reset for the default state of each digitizer.
- Sends initialization commands to set the instrument(s) to the state necessary for the operation of the instrument driver.
- Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver VI calls.

You can create sessions that include multiple instruments of the same model and in the same chassis. This automatically synchronizes all channels of the included instruments, simplifying your test program development and maintenance.


Related topics

- [NI-SCOPE Programming Flow](#)
- [NI-SCOPE Tutorial](#)



 resource name specifies the device name assigned by Measurement & Automation Explorer (MAX) to an NI-SCOPE instrument, for example, **PXI1Slot3**, where PXI1Slot3 is an instrument resource name.

This parameter accepts a comma-delimited list of strings in the form **PXI1Slot3,PXI1Slot4**, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

 **Note** You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.


Examples

Example	Device Type	Syntax
1	NI-DAQmx device	myDAQmxDevice (myDAQmxDevice = device name)
2	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::2 (2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternative syntax for NI-DAQmx devices consists of DAQ::NI-DAQmx device name, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.

You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

 **Note** NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches

the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

TF id query verifies that the digitizer you initialize is supported by NI-SCOPE. NI-SCOPE automatically performs this query, so setting this parameter is not necessary.

TF reset device specifies whether to reset the instrument during the initialization procedure.
Default Value: TRUE

Err error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

I/O instrument handle identifies a particular instrument session.

Err error out contains error information. This output provides standard error out functionality.

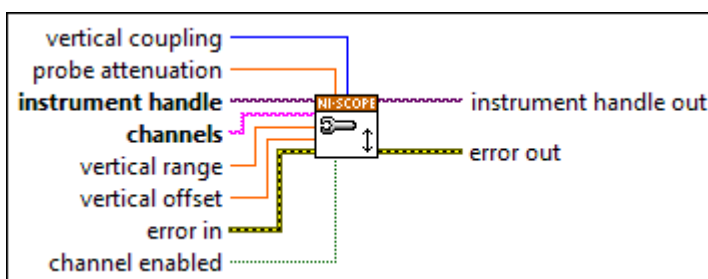
niScope Configure Vertical


Configures the most commonly configured properties of the digitizer vertical subsystem, such as the range, offset, coupling, probe attenuation, and the channel name.


Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics

- [Configuring the Vertical Settings](#)
- [NI-SCOPE Programming Flow](#)
- [Coercions of Vertical Parameters](#)




 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 vertical range specifies the absolute value of the input range for a channel.


For example, to acquire a sine wave that spans between -5 and $+5$ V, set the vertical range to 10.0 V.


 vertical offset specifies the location of the center of the range with respect to ground.


For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this property to 5.0 V.

 probe attenuation specifies the probe attenuation for the input channel.

Default Value: 1


 vertical coupling specifies how to couple the input signal. When input coupling changes, the input stage takes a finite amount of time to settle.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 channel enabled specifies whether the channel is enabled for acquisition.

Default Value: TRUE

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

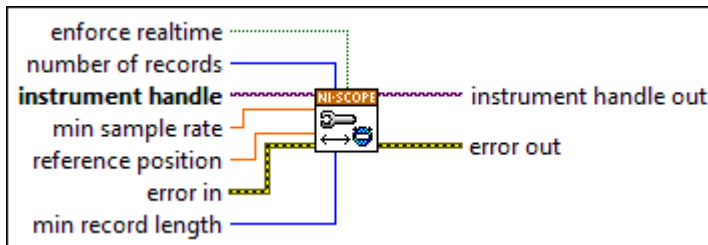
niScope Configure Horizontal Timing

Configures the common properties of the horizontal subsystem for a single record or multirecord acquisition, such as minimum sample rate and acquisition size.

Related topics

- [Configuring the Horizontal Settings](#)

- [NI-TClk Overview](#)
- [Coercions of Horizontal Parameters](#)



I/O instrument handle identifies a particular instrument session.

DBL min sample rate specifies the sample rate for the acquisition.

Default Value: 20 MS/s

DBL reference position specifies the position of the Reference Event in the waveform record as a percentage of the record.

Default Value: 50%

TF enforce realtime indicates whether the digitizer enforces real-time measurements or allows equivalent-time (RIS) measurements.

Default Value: TRUE

Defined Values

TRUE (Default)—Allow real-time acquisitions only

FALSE—Allow real-time and RIS acquisitions

Note Not all digitizers support RIS—refer to [Features Supported by Device](#) for more information.

I32 number of records specifies the number of records to acquire.


Default Value: 1


I32 min record length passes the minimum number of points you need in the record for each channel.


Call [niScope Actual Record Length](#) to obtain the actual record length used. Refer to [Coercions of Horizontal Parameters](#) for more information about why the record length may

be different than what was specified. The value must be greater than 1 and is limited by available memory.

Default Value: 1000

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Configure Trigger (poly)


Configures the digitizer for different types of triggering.

When you initiate an acquisition, the trigger system operates in the following manner:

- The digitizer waits for the start trigger, which is configured through the [Start Trigger Source](#) property. The default is VAL_IMMEDIATE.
- Upon receiving the start trigger, the digitizer begins sampling pretrigger points.
- After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a Configure Trigger instance.
- Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling.

With each Configure Trigger instance, you specify configuration parameters such as the trigger source and the amount of trigger delay. Additionally, you can adjust the amount of pretrigger and posttrigger samples using the reference position parameter in [niScope Configure Horizontal Timing](#). The default is half the record length.

For multirecord acquisitions, all records after the first record are started based on the setting of the [Advance Trigger Source](#) property. The default is VAL_IMMEDIATE.

 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

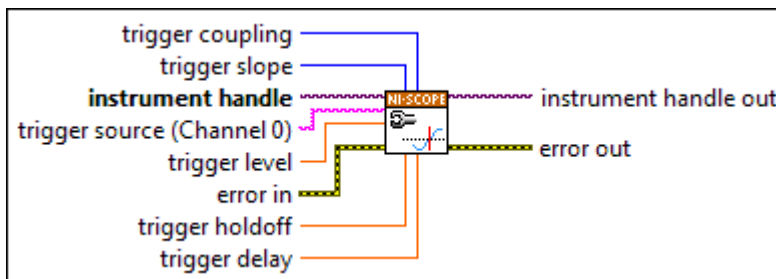
Related topics

- [Triggering](#)
- [NI-SCOPE Programming Flow](#)


Analog Edge Ref Trigger

Configures common properties for edge triggering.

An edge trigger occurs when a signal crosses a trigger threshold you specify.




 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.

 trigger level specifies the voltage threshold for the trigger.


Default Value: 0.0 V





 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to the [Trigger Slope](#) property for defined values.

 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

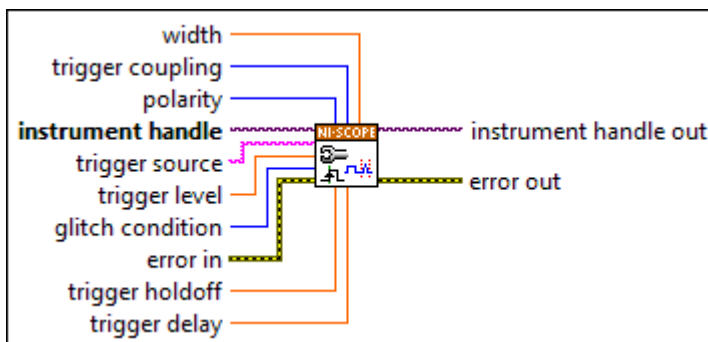
 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


-  instrument handle out has the same value as the instrument handle.
-  trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.
Default Value: 0.0 s
-  trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.
Default Value: 0.0 s
-  error out contains error information. This output provides [standard error out](#) functionality.

Analog Glitch Ref Trigger


Configures common properties for glitch triggering.

A glitch trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a width that is either greater than or less than a duration you specify.




-  width specifies, in seconds, the glitch duration to be used in combination with the glitch condition that triggers the oscilloscope.


The oscilloscope triggers when it detects a pulse of duration either less than or greater than this value depending on the value of the [Glitch Condition](#) property.


Refer to the [Glitch Width](#) property for more information.
-  trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

 polarity specifies the polarity of the pulses that trigger the oscilloscope for glitch triggering.


Refer to the [Glitch Polarity](#) property for defined values.

 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.


 trigger level specifies the voltage threshold for the trigger.

Default Value: 0.0 V


 glitch condition specifies whether the oscilloscope triggers on pulses of duration less than or greater than the specified [Glitch Width](#).

Refer to the [Glitch Condition](#) property for defined values.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

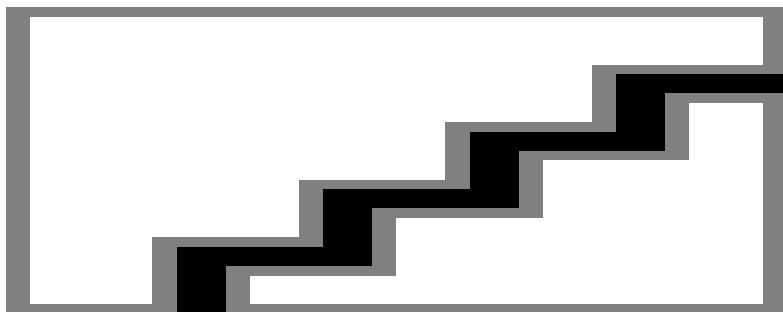
 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Analog Hysteresis Ref Trigger

Configures common properties for hysteresis triggering. This kind of trigger specifies an additional value, specified in the hysteresis parameter, that a signal

must pass through before a trigger can occur. This additional value acts as a buffer zone that keeps noise from triggering an acquisition.



I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

DBL trigger level specifies the voltage threshold for the trigger.

Default Value: 0.0 V

DBL hysteresis specifies, in volts, the size of the hysteresis window on either side of the trigger level. The digitizer triggers when the trigger signal passes through the hysteresis value you specify with this parameter, has the slope you specify with the trigger slope, and passes through the trigger level.

Default Value: 0.05 V

Valid Values


Positive trigger slope	$0 \leq \text{Hysteresis} \leq \text{Trigger Level} + (\text{Vertical Range} / 2) - \text{Vertical Offset}$
Negative trigger slope	$0 \leq \text{Hysteresis} \leq (\text{Vertical Range} / 2) + \text{Vertical Offset} - \text{Trigger Level}$


I32 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.


Refer to the [Trigger Slope](#) property for defined values.

U16 trigger coupling specifies how to couple the trigger signal.


Refer to the [Trigger Coupling](#) property for defined values.

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 **instrument handle out** has the same value as the instrument handle.

 **trigger holdoff** specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 **trigger delay** specifies how long the digitizer waits after it receives the trigger to start acquiring data.


Default Value: 0.0 s

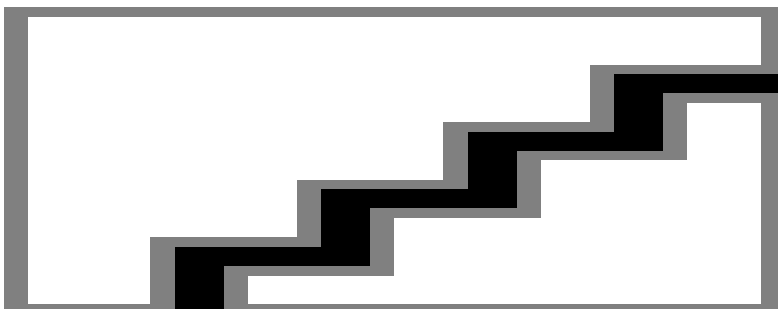
 **error out** contains error information. This output provides [standard error out](#) functionality.


Analog Runt Ref Trigger

Configures common properties for runt triggering.


A runt trigger occurs when both the leading edge and trailing edge of a pulse cross only one of two trigger thresholds you specify and with a polarity you specify, where the polarity is relative to the threshold crossed.

 **Note** You can add time qualification to trigger on runt pulses that, in addition to meeting your amplitude and polarity criteria, also meet duration criteria. Add time qualification to the runt trigger with the [Runt Time Condition](#), [Runt Time Low Limit](#), and [Runt Time High Limit](#) properties. Time qualification is disabled by default.





 **trigger coupling** specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

 polarity specifies the polarity of the runt pulses, relative to the runt threshold the pulses cross, that trigger the oscilloscope for runt triggering.


Refer to the [Runt Polarity](#) property for defined values.

 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.


 low threshold specifies, in volts, the lower of two thresholds that bound the vertical range to examine for runt pulses.

Refer to the [Runt Low Threshold](#) property for more information.


 high threshold specifies, in volts, the higher of two thresholds that bound the vertical range to examine for runt pulses.

Refer to the [Runt High Threshold](#) property for more information.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

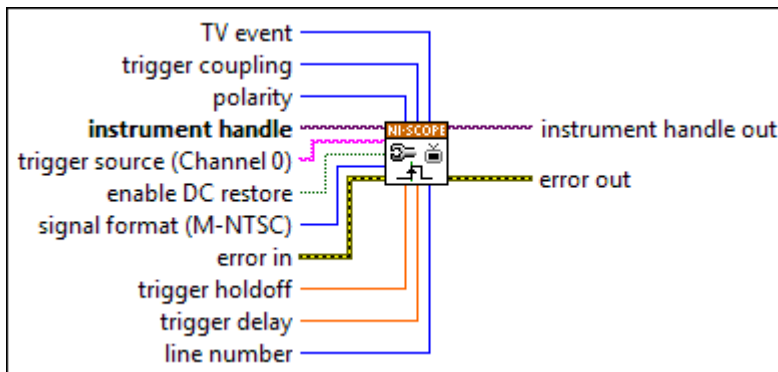
Default Value: 0.0 s

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Video Ref Trigger

Configures the common properties specific to video triggering. The video triggering properties are polarity, enable DC restore, signal format, event, and line number. A video trigger occurs when the digitizer finds a valid video signal sync. Use the trigger holdoff to skip a specific number of frames between acquisitions. For example, to acquire a specific line number multiple times and repeat the same chroma phase, skip one frame in NTSC ($121 \text{ ms} < \text{holdoff} < 159 \text{ ms}$) and five frames in SECAM ($201 \text{ ms} < \text{holdoff} < 239 \text{ ms}$).



I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

TF enable DC restore offsets each video line so the clamping level (the portion of the video line between the color burst and the beginning of the active image) is moved to zero volt.

Default Value: FALSE

I32 signal format specifies the video format to use.

Refer to the [Signal Format](#) property for defined values.


U16 polarity specifies the polarity of the video sync.

Default Value: negative


Defined Values:

positive


negative


 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.


 TV event specifies what TV event to trigger on.

Refer to the [Event](#) property for defined values.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.


Default Value: 0.0 s


 line number specifies the line number to trigger on.

The line number range covers an entire frame and is referenced as shown in [Vertical Blanking and Synchronization Signal](#).

The number of lines available depends on the format. For valid ranges, refer to the [Line Number](#) property.

Default value: 1

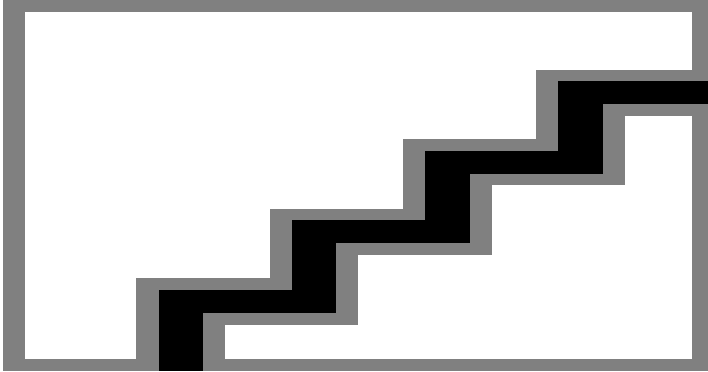
 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Analog Width Ref Trigger

Configures common properties for width triggering.

A width trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a duration that is either within or outside a duration range you specify.



DBL low threshold specifies, in seconds, the lower bound on the range of pulse durations that triggers the oscilloscope.

Refer to the [Width Low Threshold](#) property for more information.

U16 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

I32 polarity specifies the polarity of the pulses that trigger the oscilloscope for width triggering.

Refer to the [Width Polarity](#) property for defined values.

I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.


DBL trigger level specifies the voltage threshold for the trigger.

Default Value: 0.0 V


I32 width condition specifies whether the oscilloscope triggers on pulses of duration within or outside the range of pulse durations bounded by low threshold and high threshold.

Refer to the [Width Condition](#) property for defined values.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.


Default Value: 0.0 s


 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

 high threshold specifies, in seconds, the upper bound on the range of pulse durations that triggers the oscilloscope.

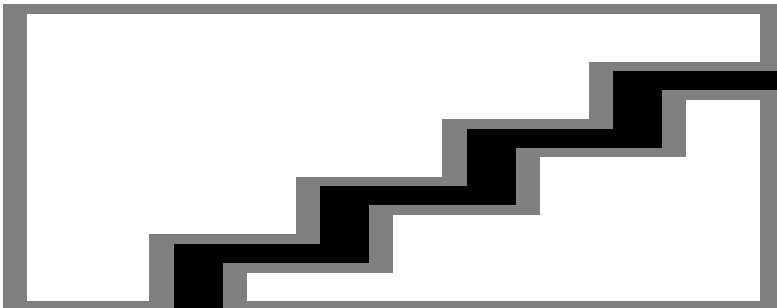
Refer to the [Width High Threshold](#) property for more information.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Analog Window Ref Trigger

Configures common properties for window triggering. A window trigger occurs when a signal enters or leaves a window you specify with the high level or low level parameters.




 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.

 low level passes the voltage threshold for the digitizer to use for low triggering.


Default Value: 0 V

 high level passes the voltage threshold for the digitizer to use for high triggering.


Default Value: 0.10 V


 trigger window mode specifies whether the trigger should occur when the signal is entering or leaving a window.

Default Value: Entering


 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

 instrument handle out has the same value as the instrument handle.

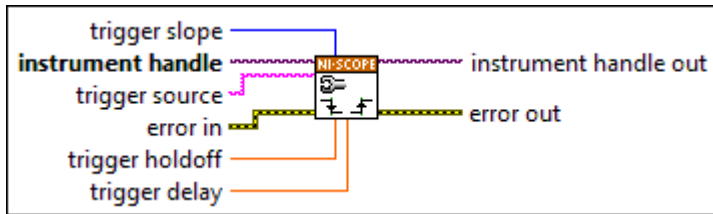
 error out contains error information. This output provides [standard error out](#) functionality.

Digital Edge Ref Trigger

Configures the common properties of a digital trigger.



Note Digital triggering is not supported in [RIS mode](#).



I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

I32 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to the [Trigger Slope](#) property for defined values.

Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

I/O instrument handle out has the same value as the instrument handle.

Err error out contains error information. This output provides [standard error out](#) functionality.

Immediate Ref Trigger

Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.

Related topics:

- [Immediate Triggers](#)



I/O instrument handle identifies a particular instrument session.

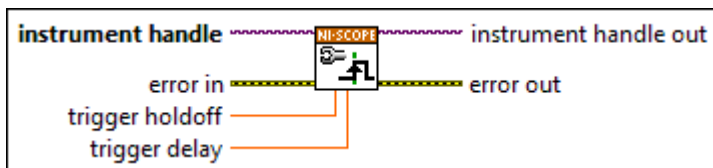
FE error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

FE error out contains error information. This output provides [standard error out](#) functionality.

Software Ref Trigger

Configures common properties for software triggering. This VI only works in an Initiate/Fetch operation.



I/O instrument handle identifies a particular instrument session.

FE error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

FE error out contains error information. This output provides [standard error out](#) functionality.

niScope Read (poly)

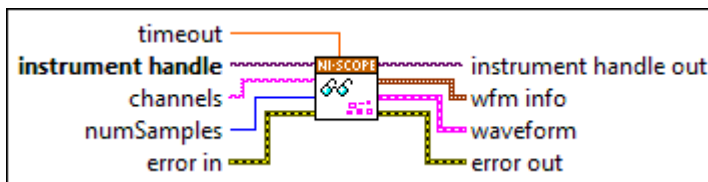
Initiates an acquisition, waits for it to complete, and acquires data.

Related topics:

- [Acquisition Functions](#)
- [NI-SCOPE Programming Flow](#)

Cluster

Initiates an acquisition, waits for it to complete, and retrieves the data for a single channel and record.



I/O instrument handle identifies a particular instrument session.


abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


I/O instrument handle out has the same value as the instrument handle.


 wfmmInfo contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 waveform is a cluster containing the initial x value, xincrement, and a waveform array. This output can be wired directly to the LabVIEW waveform graph, so the waveform is plotted with timing information. The initial x value in the cluster is relative to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

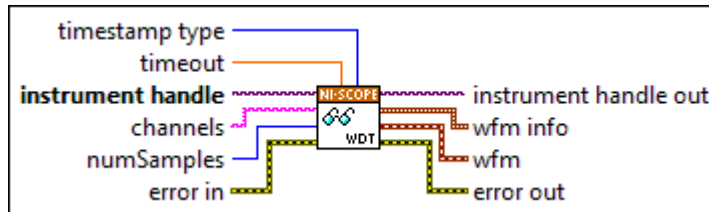
DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL wfm is an array of waveform data containing one waveform without any timing information.

Panel error out contains error information. This output provides [standard error out](#) functionality.

WDT

Returns the waveform the digitizer acquires for the specified channel. The VI initiates an acquisition that returns a scaled voltage waveform in a waveform data type that includes timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


I32 timestamp Type specifies the time basis for the timestamp on the WDT data.


Defined Values


absolute


relative


 error in describes error conditions that occur before this node runs. This input provides standard error in functionality.


 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage = binary\ data \times gain\ factor + offset}$$

 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\mathbf{voltage = binary\ data \times gain\ factor + offset}$$

 reserved1 is reserved. Do not use.

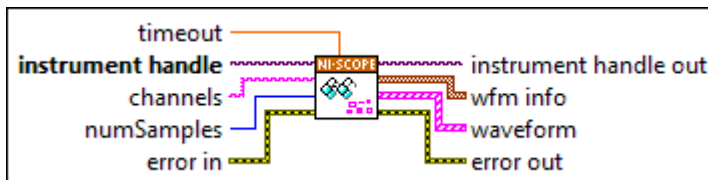
 reserved2 is reserved. Do not use.

 wfm returns data for wiring to a graph, including the timing information.


 error out contains error information. This output provides [standard error out](#) functionality.


1D Cluster


Initiates an acquisition, waits for it to complete, and retrieves the data for multiple channels and records.





 instrument handle identifies a particular instrument session.

 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

Waveform waveform is an array of clusters, each containing the initial x value, x increment, and a waveform array. This output can be wired directly to the LabVIEW waveform graph, so each waveform in the array is plotted with timing information. The initial x value in the cluster is the value relative to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

The array of clusters includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal there is one waveform per channel per record. If you call the fetch VI during a normal acquisition with the channel string "0,1" the order of the output is:

record 0, channel 0

record 0, channel 1

record 1, channel 0

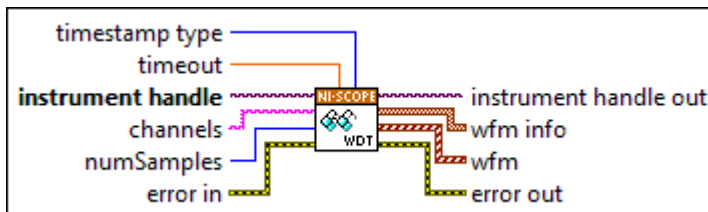
record 1, channel 1

- DBL** relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.
- DBL** xIncrement indicates the time in seconds between two samples in the acquired waveform.
- DBL** wfm is an array of waveform data containing one waveform without any timing information.

Err error out contains error information. This output provides [standard error out](#) functionality.

1D WDT

Initiates an acquisition and returns a one-dimensional array of LabVIEW waveform data types that includes timing information. This VI is only supported in LabVIEW 7.0 or later.




I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 timestamp Type specifies the time basis for the timestamp on the WDT data.


Defined Values


absolute


relative


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 `wfm` returns a one-dimensional array of LabVIEW waveform data types that contain timing information. If you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.


For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

index 2 = record 1, channel 0

index 3 = record 1, channel 1

 `error out` contains error information. This output provides [standard error out](#) functionality.

niScope Close

Performs the following actions:

- Closes the instrument I/O session.
- Destroys the IVI session and all of its properties.
- Deallocates any memory resources used by the IVI session.

Call this VI when you are finished using an instrument driver session.

Related topics

- [NI-SCOPE Programming Flow](#)




 `instrument handle` identifies a particular instrument session.

 `error in` describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Initialize With Options

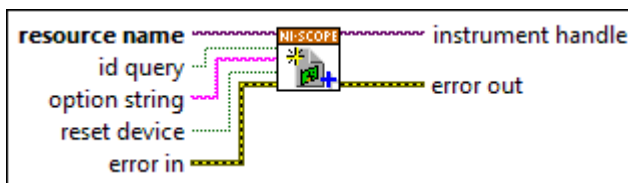
Performs the following initialization actions:


- Creates a new IVI instrument driver session and optionally sets the initial state of the following session properties: Range Check, Cache, Simulate, Record Value Coercions.
- Opens a session to the device(s) that you specify in the resource name parameter.
- Queries each instrument ID and verifies that it is valid for this instrument driver.
- Resets the instrument(s) to a known state if the reset device parameter is set to TRUE; refer to niScope Reset for the default state of each digitizer.
- Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver VI calls.

You can create sessions that include multiple instruments of the same model and in the same chassis. This automatically synchronizes all channels of the included instruments, simplifying your test program development and maintenance.


Related topics

- [NI-SCOPE Programming Flow](#)
- [NI-SCOPE Tutorial](#)



 resource name specifies the device name assigned by Measurement & Automation Explorer (MAX) to an NI-SCOPE instrument, for example, **PXI1Slot3**, where PXI1Slot3 is an instrument resource name.

This parameter accepts a comma-delimited list of strings in the form **PXI1Slot3,PXI1Slot4**, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

 **Note** You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.


Examples


Example	Device Type	Syntax
1	NI-DAQmx device	myDAQmxDevice (myDAQmx Device = device name)
2	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::2 (2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternative syntax for NI-DAQmx devices consists of DAQ::NI-DAQmx device name, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.

You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

 **Note** NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

 **TF** id query verifies that the digitizer you initialize is supported by NI-SCOPE. NI-SCOPE automatically performs this query, so setting this parameter is not necessary.

abc option string sets the initial value of certain properties for the session.

The following table lists the properties and the name you use in this parameter to identify the property.

Name	Attribute Defined Constant	Default Value
RangeCheck	niScope»Inherent IVI Settings »User Options»Range Check	TRUE
Cache	niScope»Inherent IVI Settings »User Options»Cache	TRUE
Simulate	niScope»Inherent IVI Settings »User Options»Simulate	FALSE
RecordCoercions	niScope»Inherent IVI Settings »User Options»Record Value Coercions	FALSE
QueryInstrStatus	niScope»Inherent IVI Settings »User Options»Query Instrum ent Status	TRUE

Default Values: "Simulate=0,RangeCheck=1,Cache=1"

You can use the `DriverSetup` flag to simulate a device, attach an accessory to your device session, or load a bitfile to a device FPGA.


- To simulate a device, specify the model and board type you wish to simulate. For example, `Simulate = 1, DriverSetup = Model:5122; BoardType:PX I` will simulate a NI 5122.
- To attach an accessory to your device session, specify the name given to the accessory by MAX. For example, `DriverSetup = Accessory:Dev1` will attach the accessory named "Dev1" in MAX to your device session.
- For instructions on how to load a bitfile to a device FPGA, refer to [Using NI-SCOPE Instrument Driver FPGA Extensions](#).


For more information about simulation, refer to the `niScope EX Simulated Acquisition` example.


To see this parameter used in a VI, refer to the `niScope EX External Amplifier` example.

TF `reset device` specifies whether to reset the instrument during the initialization procedure.

Default Value: TRUE

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle identifies a particular instrument session.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Property Node

The niScope Property Node is used to set, get, or check properties.



Configuration

Owning Palette: [NI-SCOPE VIs](#)


Use the NI-SCOPE Configuration VIs to set up and transfer the parameters of your acquisition, or use Auto Setup to automatically configure device settings.

Palette Object	Description
niScope Configure Vertical	Configures the most commonly configured properties of the digitizer vertical subsystem, such as the range, offset, coupling, probe attenuation, and the channel name.
niScope Auto Setup	Automatically configures the digitizer.
niScope Configure Acquisition	Configures how the oscilloscope acquires data and fills the waveform record.
niScope Configure Chan Characteristics	Configures the properties that control the electrical characteristics of the channel. These properties are the input impedance and the bandwidth.
niScope Export Attribute Configuration (poly)	Exports the attribute configuration of a session to either a file or a buffer.

niScope Import Attribute Configuration (poly)	Imports an attribute configuration to the session from either a file or a buffer.
Subpalette	Description
Configure Onboard Signal Processing	Use the VIs on this palette to configure Onboard Signal Processing (OSP) for devices that support OSP.

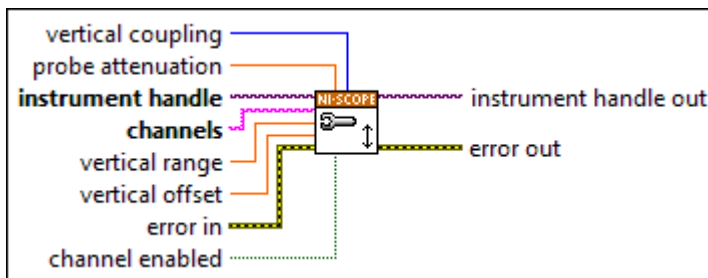
niScope Configure Vertical

Configures the most commonly configured properties of the digitizer vertical subsystem, such as the range, offset, coupling, probe attenuation, and the channel name.


 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics

- [Configuring the Vertical Settings](#)
- [NI-SCOPE Programming Flow](#)
- [Coercions of Vertical Parameters](#)




 instrument handle identifies a particular instrument session.

 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 vertical range specifies the absolute value of the input range for a channel.


For example, to acquire a sine wave that spans between -5 and $+5$ V, set the vertical range to 10.0 V.


 vertical offset specifies the location of the center of the range with respect to ground.


For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this property to 5.0 V.

 probe attenuation specifies the probe attenuation for the input channel.


Default Value: 1


 vertical coupling specifies how to couple the input signal. When input coupling changes, the input stage takes a finite amount of time to settle.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 channel enabled specifies whether the channel is enabled for acquisition.

Default Value: TRUE

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Auto Setup

Automatically configures the digitizer.

When you call this VI, the digitizer senses the input signal and automatically configures many of the instrument settings. If a signal is detected on a channel, the driver chooses the smallest available vertical range that is larger than the signal range. For example, if the signal is a 1.2 Vpk-pk sine wave, and the device supports 1 V and 2 V vertical ranges, the driver will choose the 2 V vertical range for that channel.

If no signal is found on any analog input channel, a warning is returned and all channels are enabled. A channel is considered to have a signal present if the signal is at least 10% of the smallest vertical range available for that channel.


Related topics


- [Configuring an Acquisition using Auto_Setup](#)


Details



 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Details

Settings Changed by Auto Setup

This VI changes the following settings to the following values:

General	
Acquisition mode	Normal
Reference clock	Internal
Vertical	
Vertical coupling	AC (when AC is supported; otherwise DC)
Vertical bandwidth	Full
Vertical range	Changed by Auto Setup
Vertical offset	0 V
Probe attenuation	Unchanged by Auto Setup
Input impedance	Unchanged by Auto Setup
Horizontal	
Sample rate	Changed by Auto Setup
Min record length	Changed by Auto Setup
Enforce realtime	True
Number of Records	Changed to 1
Triggering	
Trigger type	Edge if signal present, otherwise immediate

Trigger channel	Lowest numbered channel with a signal present
Trigger slope	Positive
Trigger coupling	DC
Reference position	50%
Trigger level	50% of signal on trigger channel
Trigger delay	0
Trigger holdoff	0
Trigger output	None

PXIe-5185 and PXIe-5186 Unique Behavior

Because PXIe-5185 and PXIe-5186 modules have two physical connectors (50 ohm input impedance, 1 Mohm input impedance) for each channel, this VI has unique behavior for these modules.

When you use niScope Properties to specify the input impedance on either channel or both channels, this VI searches for a signal on only that input connector. If no signal is detected, NI-SCOPE returns a warning and both channels are enabled for the input impedance specified.

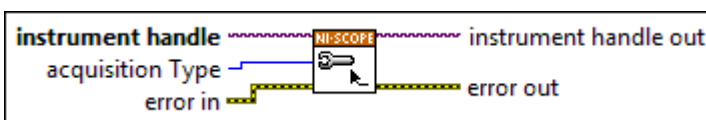
If you do not specify an input impedance, this VI searches for a signal on both input connectors (first on the 50 ohm input connector, then on the 1 Mohm input connector). If NI-SCOPE detects a signal on the 50 ohm connector (default) then it uses that signal. If NI-SCOPE does not detect a signal on the 50 ohm connector, it then searches the 1 Mohm connector for a signal. If a signal is not detected on any connectors, NI-SCOPE returns a warning and both channels are enabled with 50 ohm impedance.


niScope Configure Acquisition


Configures how the oscilloscope acquires data and fills the waveform record.

Related topics

- [Acquisition Functions](#)



 instrument handle identifies a particular instrument session.


 acquisition Type is the manner in which the oscilloscope acquires data and fills the waveform record.


Not all oscilloscopes support all acquisition types; refer to [Features Supported by Device](#) for more information.


Default Value: Normal

Defined Values

Normal (0)	Sets the oscilloscope to normal resolution mode. The oscilloscope can use real-time sampling or equivalent-time sampling.
Flex Res (1)	Sets legacy oscilloscopes to flexible resolution mode, if supported.
DDC (3)	Sets legacy oscilloscopes to DDC mode, if supported. To use DDC mode for the PXI/PCI-5142 or PXIe-5622, set this parameter to Normal and set the DDC Enabled property to True.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

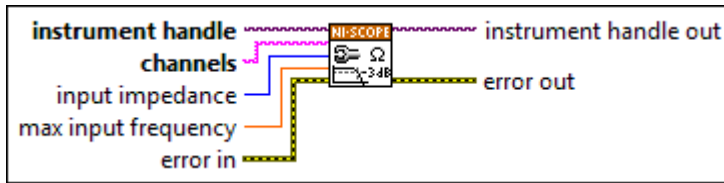
 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

niScope Configure Chan Characteristics

Configures the properties that control the electrical characteristics of the channel. These properties are the input impedance and the bandwidth.

Related topics:

- [Using Configure Chan Characteristics](#)
- [Impedance and Impedance Matching](#)



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

U32 input impedance is the input impedance for the channel.

Default Value: 1 mega ohm

Defined Values

1 mega ohm

50 ohms

DBL maximum input frequency specifies the bandwidth of the channel at which the input circuitry attenuates the signal by 3 dB. Pass 0 for this value to use the hardware default bandwidth. Pass -1 for this value to achieve full bandwidth.

Default Value: 0.00 Hz

Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

Err error out contains error information. This output provides [standard error out](#) functionality.

niScope Export Attribute Configuration (poly)

Exports the attribute configuration of a session to either a file or a buffer.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

This VI verifies that the attributes you have configured for the session are valid. If the configuration is invalid, NI-SCOPE returns an error.

Related topics:

[Attributes and Attribute Functions](#)

[Setting Attributes Before Reading Attributes](#)


[Details](#)

niScope Export Attribute Configuration File

Exports the attribute configuration of the session to the specified file.




 instrument handle identifies a particular instrument session.


 file path is the absolute path to a placeholder file you must create to contain the attribute configuration you want to export.

If you specify an empty or relative path, this VI returns an error.

Default file extension: `.niscopconfig`

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.






 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Export Attribute Configuration Buffer

Exports the attribute configuration of the session to a buffer.



-  instrument handle identifies a particular instrument session.
-  error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
-  instrument handle out has the same value as the instrument handle.
-  buffer is a byte array that contains the exported attribute configuration.
-  error out contains error information. This output provides [standard error out](#) functionality.


Details

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the resource name input to the [niScope Initialize With Options](#) or [niScope Initialize VIs](#).

For example, if your entry for resource name is PXI1Slot1, PXI1Slot2 for the exporting session and PXI2Slot2, PXI2Slot3 for the importing session:


- The configuration exported from PXI1Slot1 is imported into PXI2Slot2.
- The configuration exported from PXI1Slot2 is imported into PXI2Slot3.

 Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

niScope Import Attribute Configuration (poly)

Imports an attribute configuration to the session from either a file or a buffer.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

 Note You cannot call this VI while the session is in a running state, such as while acquiring a signal.

Related topics:

[Attributes and Attribute Functions](#)

[Setting Attributes Before Reading Attributes](#)


[Details](#)

niScope Import Attribute Configuration File

Imports an attribute configuration to the session from the specified file.




 **instrument handle** identifies a particular instrument session.


 **file path** is the absolute path to the file that contains the attribute configuration to import.

If you specify an empty or relative path, this VI returns an error.

Default file extension: `.niscopescopeconfig`

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **instrument handle out** has the same value as the instrument handle.


 **error out** contains error information. This output provides [standard error out](#) functionality.


niScope Import Attribute Configuration Buffer


Imports an attribute configuration to the session from the specified buffer.




 **instrument handle** identifies a particular instrument session.

 `buffer` is a byte array that contains the attribute configuration to import.

 `error in` describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 `instrument handle out` has the same value as the `instrument handle`.

 `error out` contains error information. This output provides [standard error out](#) functionality.


Details

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the resource name input to the [niScope Initialize With Options](#) or [niScope Initialize VIs](#).

For example, if your entry for resource name is `PXI1Slot1`, `PXI1Slot2` for the exporting session and `PXI2Slot2`, `PXI2Slot3` for the importing session:

- The configuration exported from `PXI1Slot1` is imported into `PXI2Slot2`.
- The configuration exported from `PXI1Slot2` is imported into `PXI2Slot3`.

 Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

Configure Onboard Signal Processing

Owning Palette: [Configuration](#)


Use the VIs on this palette to configure Onboard Signal Processing (OSP) for devices that support OSP.

PaletteObject	Description
niScope Configure Equalization Filter Coefficients	Configures the custom coefficients for the equalization FIR filter on the device. This filter is designed to compensate the input signal for artifacts




	introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.
niScope Get Equalization Filter Coefficients	Retrieves the custom coefficients for the equalization FIR filter on the device.
niScope Get Frequency Response	Gets the frequency response of the digitizer for the current configurations of the channel attributes. This VI can be used only with high-speed digitizers that support onboard signal processing (OSP).


niScope Configure Equalization Filter Coefficients


Configures the custom coefficients for the equalization FIR filter on the device. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.


 Note This VI can be used only with high-speed digitizers that support onboard signal processing (OSP).



-  instrument handle identifies a particular instrument session.
-  channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).
-  coefficients are the custom coefficients for the equalization FIR filter on the device. These coefficients should be between +1 and -1. You can obtain the number of coefficients from the [Equalization Num Coefficients](#) property. The [Equalization Filter Enabled](#) property must be set to TRUE to enable the filter.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

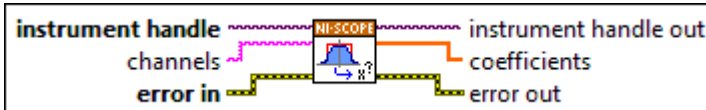
 error out contains error information. This output provides [standard error out](#) functionality.

niScope Get Equalization Filter Coefficients


Retrieves the custom coefficients for the equalization FIR filter on the device.


This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the device. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.


 Note This VI can be used only with high-speed digitizers that support onboard signal processing (OSP).



 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

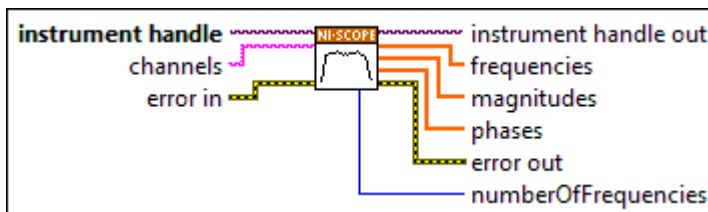
 coefficients are the custom coefficients for the equalization FIR filter on the device.

These coefficients should be between +1 and -1. You can obtain the number of coefficients from the [Equalization Num Coefficients](#) property. The [Equalization Filter Enabled](#) property must be set to TRUE to enable the filter.


 error out contains error information. This output provides [standard error out](#) functionality.


niScope Get Frequency Response


Gets the frequency response of the digitizer for the current configurations of the channel attributes. This VI can be used only with high-speed digitizers that support onboard signal processing (OSP).





 **instrument handle** identifies a particular instrument session.

 **channels** specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 **instrument handle out** has the same value as the **instrument handle**.

 **frequencies** is an array of frequencies that corresponds with the amplitude and phase response of the device.

 **magnitudes** is the array of magnitudes that correspond with the magnitude response of the device.

 **phases** is the array of phases that correspond with the phase response of the device.

 **numberOfFrequencies** returns the number of frequencies in the returned spectrum.

 **error out** contains error information. This output provides [standard error out](#) functionality.

Timing

Owning Palette: [NI-SCOPE VIs](#)

Use the NI-SCOPE Timing VIs to configure common timing properties for your digitizer.

Palette Object	Description
niScope Configure Horizontal Timing	Configures the common properties of the horizontal subsystem for a single record or multirecord acquisition, such as minimum sample rate and acquisition size.
niScope Configure Clock	Configures the properties for synchronizing the digitizer to an external clock or sending the digitizer's clock output to be used as a synchronizing clock for other devices.
niScope Export Signal	Configures the digitizer to generate a signal that other devices can detect when configured for digital triggering or sharing clocks. The signal parameter specifies what condition causes the digitizer to generate the signal. The output terminal parameter specifies where to send the signal on the hardware (such as a PFI connector or RTSI line).
niScope Adjust Sample Clock Relative Delay	Applies offset, in seconds, to the sample clock relative to the reference clock when using the on board clock.
Subpalette	Description
Actual Values	Use the VIs on the Actual Values palette to configure actual sample mode, record length, and sample rate.

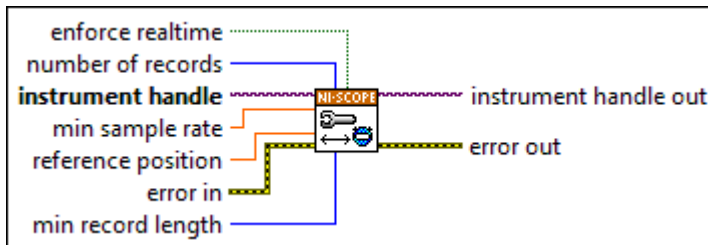
niScope Configure Horizontal Timing

Configures the common properties of the horizontal subsystem for a single record or multirecord acquisition, such as minimum sample rate and acquisition size.

Related topics

- [Configuring the Horizontal Settings](#)

- [NI-TClk Overview](#)
- [Coercions of Horizontal Parameters](#)



I/O instrument handle identifies a particular instrument session.

DBL min sample rate specifies the sample rate for the acquisition.

Default Value: 20 MS/s

DBL reference position specifies the position of the Reference Event in the waveform record as a percentage of the record.

Default Value: 50%

TF enforce realtime indicates whether the digitizer enforces real-time measurements or allows equivalent-time (RIS) measurements.

Default Value: TRUE

Defined Values

TRUE (Default)—Allow real-time acquisitions only

FALSE—Allow real-time and RIS acquisitions

Note Not all digitizers support RIS—refer to [Features Supported by Device](#) for more information.

I32 number of records specifies the number of records to acquire.


Default Value: 1


I32 min record length passes the minimum number of points you need in the record for each channel.


Call [niScope Actual Record Length](#) to obtain the actual record length used. Refer to [Coercions of Horizontal Parameters](#) for more information about why the record length may

be different than what was specified. The value must be greater than 1 and is limited by available memory.

Default Value: 1000


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

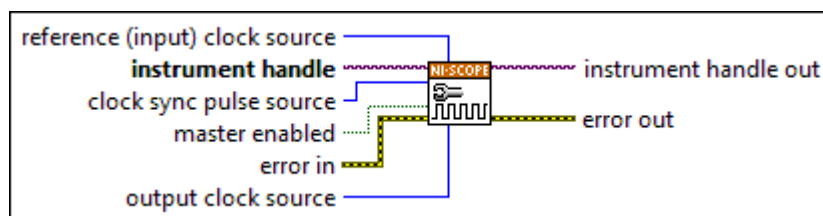
niScope Configure Clock

Configures the properties for synchronizing the digitizer to an external clock or sending the digitizer's clock output to be used as a synchronizing clock for other devices.


 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics

- [Sample Clock](#)
- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)




 instrument handle identifies a particular instrument session.

 clock sync pulse source specifies the line on which the sample clock or the one-time sync pulse is sent or received.


Default Value: RTSI 0

Refer to the [Clock Sync Pulse Source](#) property for defined values.


 **master enabled** specifies whether the device is a master or a slave; the master device is typically the originator of the trigger signal and clock sync pulse.

For a standalone device, set this parameter to FALSE.


Default Value: FALSE

 **input clock source** specifies the input source for the PLL reference clock (such as the 1-20 MHz clock on SMC-based devices) to which the digitizer is phase-locked for all digitizers. Refer to the [Reference \(Input\) Clock Source](#) property for defined values.


Default Value: None

 **output clock source** specifies the output source for the PLL reference clock to which the sample clock of another digitizer can be phase-locked. Refer to the [Output Clock Source](#) property for more information.

Default Value: None

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **instrument handle out** has the same value as the instrument handle.

 **error out** contains error information. This output provides [standard error out](#) functionality.

niScope Export Signal

Configures the digitizer to generate a signal that other devices can detect when configured for digital triggering or sharing clocks. The signal parameter specifies what condition causes the digitizer to generate the signal. The output terminal parameter specifies where to send the signal on the hardware (such as a PFI connector or RTSI line).

In cases where multiple instances of a particular signal exist, use the signal identifier input to specify which instance to control. For normal events, only one instance exists and you should leave signal identifier set to None. You can call this VI multiple times, and set each line available to a different signal.

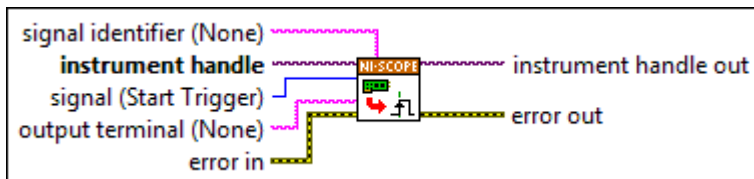
To unprogram a specific line on a device, call this VI with the signal you no longer want to export and set output terminal to None.



Note This VI replaces [niScope Configure Trigger Output](#).

Related topics:

- [Triggering](#)
- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [PXI Trigger Lines](#)



I/O instrument handle identifies a particular instrument session.

UI6 signal specifies the signal (clock, trigger, or event) to export.

Defined Values

None (0)—Do not generate a digital pulse.

Reference Trigger (1)—Generate a pulse when detecting the stop/reference trigger.

Start Trigger (2)—Generate a pulse at the start of the acquisition.

End of Acquisition (3)—Generate a pulse at the end of the acquisition.

End of Record (4)—Generate a pulse at the end of each record.

Record Advance (5)—Generate a pulse when the digitizer is advancing to the next record.

Ready for Record Advance (6)—Asserts when the digitizer is ready to advance to the next record.

Ready for Start (7)—Asserts when the digitizer is initiated and ready to accept a start trigger to begin sampling.

Ready for Reference Trigger (10)—Asserts when the digitizer is ready to accept a reference trigger.

Reference Clock (100)—Export the reference clock for the digitizer to the specified terminal.

Sample Clock (101)—Export the sample clock for the digitizer to the specified terminal.

5 Volt Power (13)—Export a 5 V power source.

 output terminal identifies the hardware signal line on which the digital pulse is generated.

Defined Values

None (default)

PXI Trigger Line 0/RTSI 0

PXI Trigger Line 1/RTSI 1

PXI Trigger Line 2/RTSI 2

PXI Trigger Line 3/RTSI 3

PXI Trigger Line 4/RTSI 4

PXI Trigger Line 5/RTSI 5

PXI Trigger Line 6/RTSI 6

PXI Trigger Line 7/RTSI 7 (RTSI Clock)

PXI Star Trigger

PFI 0

PFI 1

PFI 2

PFI 3

PFI 4


PFI 5


PFI 6

PFI 7

Clock Out

 signal identifier describes the signal being exported.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Adjust Sample Clock Relative Delay

Applies offset, in seconds, to the sample clock relative to the reference clock when using the onboard clock.

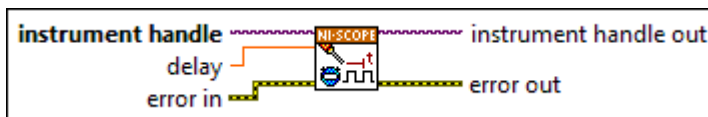
Each time this VI is called, the sample clock is offset from the reference clock by the specified amount of time.

Adjustment range: ± 1 **Sample Clock Period** per call

Related topics:

- [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#)
- [Sample Clock](#)
- [Reference Clock/Phase-Lock Loop](#)
- [NI-TClk Overview](#)

Details



I/O instrument handle identifies a particular instrument session.

DBL delay is the amount of time (in seconds) to delay the sample clock. This value is relative, so repeated calls to this VI delay the sample clock by this amount every time.

Default Value: None

E-UI error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

E-UI error out contains error information. This output provides [standard error out](#) functionality.

Details

Programming Patterns

Use this VI when performing manual adjustment to correct for skew and jitter between oscilloscopes synchronized with NI-TClk.

1. Apply time offset with this VI to reduce skew and/or jitter.
2. Use the [Oscillator Phase DAC Value](#) property to convert this time offset into a value that can apply the manual adjustment across sessions and improve synchronization repeatability.

For details on performing manual adjustment, refer to [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#).

Actual Values

Owning Palette: [Timing](#)

Use the VIs on the Actual Values palette to configure actual sample mode, record length, and sample rate.

Palette Object	Description
niScope Actual Sample Mode	Returns the sample mode the digitizer is currently using.
niScope Actual Record Length	Returns the actual number of points the digitizer acquires for each channel. After configuring the digitizer for an acquisition, call this VI to determine the size of the waveforms that the digitizer acquires. The value is equal to or greater than the minimum number of points specified in any of the Configure Horizontal VIs.
niScope Sample Rate	Returns the effective sample rate, in samples per second, of the acquired waveform using the current configuration.

niScope Actual Sample Mode

Returns the sample mode the digitizer is currently using.

Related topics:

- [Sampling Methods](#)



I/O instrument handle identifies a particular instrument session.

Error error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

Integer sample mode returns the sample mode the digitizer is currently using.

Error error out contains error information. This output provides [standard error out](#) functionality.

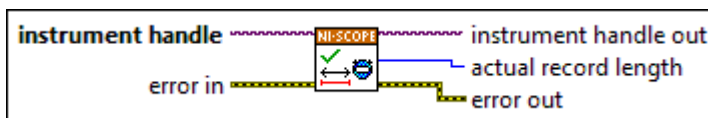
niScope Actual Record Length


Returns the actual number of points the digitizer acquires for each channel. After configuring the digitizer for an acquisition, call this VI to determine the size of the waveforms that the digitizer acquires. The value is equal to or greater than the minimum number of points specified in any of the Configure Horizontal VIs.


Use the record length returned by this VI as the input to the numSamples parameter of the Read and Fetch VIs.


Related topics


- [Record Length](#)
- [Coercions of Horizontal Parameters](#)
- [Making Multiple Record Acquisitions](#)
- [Acquiring Data Continuously](#)




 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 actual record length returns the actual number of points the digitizer acquires for each channel.

NI-SCOPE returns the value held in the [Horizontal Actual Record Length](#) property; refer to [Coercions of Horizontal Parameters](#) for more information.

 error out contains error information. This output provides [standard error out](#) functionality.

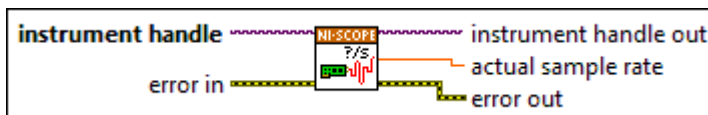
niScope Sample Rate

Returns the effective sample rate, in samples per second, of the acquired waveform using the current configuration.


Refer to [Coercions of Horizontal Parameters](#) for more information about sample rate coercion.


Related topics


- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)
- [Sample Clock](#)




 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 actual sample rate returns the effective sample rate of the acquired waveform the digitizer acquires for each channel.

 error out contains error information. This output provides [standard error out](#) functionality.

Triggering

Owning Palette: [NI-SCOPE VIs](#)

Use the NI-SCOPE Triggering VIs to configure the trigger properties.

PaletteObject	Description
niScope Configure Trigger (poly)	Configures the digitizer for different types of triggering.
niScope Export Signal	Configures the digitizer to generate a signal that other devices can detect when configured for digital triggering or sharing clocks. The signal parameter specifies what condition causes the digitizer to generate the signal. The output terminal parameter specifies where to send the signal on the hardware (such as a PFI connector or RTSI line).
niScope Send Software Trigger Edge	Sends the selected trigger to the digitizer. If you called niScope Configure Trigger Software, call this VI when you want the reference trigger to occur. You can also call this VI to override a misused edge, digital, or hysteresis reference trigger. If you have configured an Acquisition Arm Source, an Arm Reference Trigger Source, or an Advance Trigger Source, call this VI when you want to send the corresponding trigger to the digitizer.

niScope Configure Trigger (poly)


Configures the digitizer for different types of triggering.

When you initiate an acquisition, the trigger system operates in the following manner:

- The digitizer waits for the start trigger, which is configured through the [Start Trigger Source](#) property. The default is VAL_IMMEDIATE.
- Upon receiving the start trigger, the digitizer begins sampling pretrigger points.
- After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a Configure Trigger instance.
- Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling.

With each Configure Trigger instance, you specify configuration parameters such as the trigger source and the amount of trigger delay. Additionally, you can adjust the amount of pretrigger and posttrigger samples using the reference position parameter in [niScope Configure Horizontal Timing](#). The default is half the record length.

For multirecord acquisitions, all records after the first record are started based on the setting of the [Advance Trigger Source](#) property. The default is VAL_IMMEDIATE.

 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

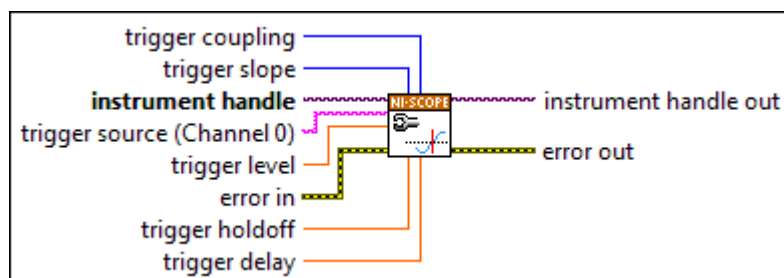
Related topics


- [Triggering](#)
- [NI-SCOPE Programming Flow](#)


Analog Edge Ref Trigger

Configures common properties for edge triggering.

An edge trigger occurs when a signal crosses a trigger threshold you specify.




 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.

 trigger level specifies the voltage threshold for the trigger.


Default Value: 0.0 V


 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.


Refer to the [Trigger Slope](#) property for defined values.

 trigger coupling specifies how to couple the trigger signal.


Refer to the [Trigger Coupling](#) property for defined values.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

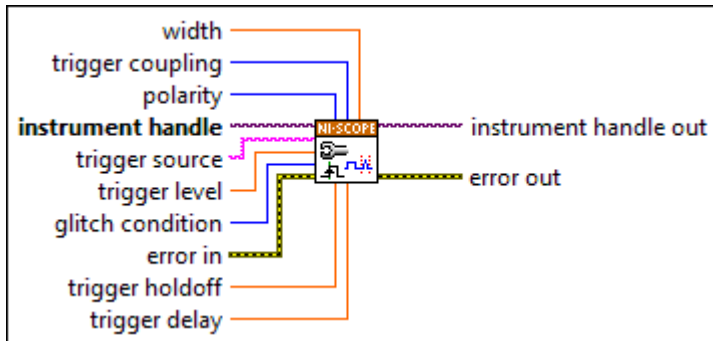
Default Value: 0.0 s

 error out contains error information. This output provides [standard error out](#) functionality.

Analog Glitch Ref Trigger

Configures common properties for glitch triggering.

A glitch trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a width that is either greater than or less than a duration you specify.



DBL width specifies, in seconds, the glitch duration to be used in combination with the glitch condition that triggers the oscilloscope.

The oscilloscope triggers when it detects a pulse of duration either less than or greater than this value depending on the value of the [Glitch Condition](#) property.

Refer to the [Glitch Width](#) property for more information.

U16 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

I32 polarity specifies the polarity of the pulses that trigger the oscilloscope for glitch triggering.

Refer to the [Glitch Polarity](#) property for defined values.

I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

DBL trigger level specifies the voltage threshold for the trigger.

Default Value: 0.0 V

I32 glitch condition specifies whether the oscilloscope triggers on pulses of duration less than or greater than the specified [Glitch Width](#).

Refer to the [Glitch Condition](#) property for defined values.

Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

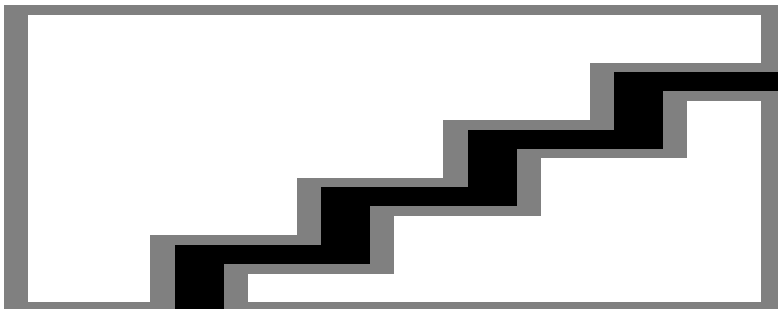
Default Value: 0.0 s

I/O instrument handle out has the same value as the instrument handle.

ERR error out contains error information. This output provides [standard error out](#) functionality.

Analog Hysteresis Ref Trigger

Configures common properties for hysteresis triggering. This kind of trigger specifies an additional value, specified in the hysteresis parameter, that a signal must pass through before a trigger can occur. This additional value acts as a buffer zone that keeps noise from triggering an acquisition.



I/O instrument handle identifies a particular instrument session.


abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

DBL trigger level specifies the voltage threshold for the trigger.


Default Value: 0.0 V

 hysteresis specifies, in volts, the size of the hysteresis window on either side of the trigger level. The digitizer triggers when the trigger signal passes through the hysteresis value you specify with this parameter, has the slope you specify with the trigger slope, and passes through the trigger level.


Default Value: 0.05 V

Valid Values


Positive trigger slope	$0 \leq \text{Hysteresis} \leq \text{Trigger Level} + (\text{Vertical Range} / 2) - \text{Vertical Offset}$
Negative trigger slope	$0 \leq \text{Hysteresis} \leq (\text{Vertical Range} / 2) + \text{Vertical Offset} - \text{Trigger Level}$


 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.


Refer to the [Trigger Slope](#) property for defined values.

 trigger coupling specifies how to couple the trigger signal.


Refer to the [Trigger Coupling](#) property for defined values.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.


Default Value: 0.0 s

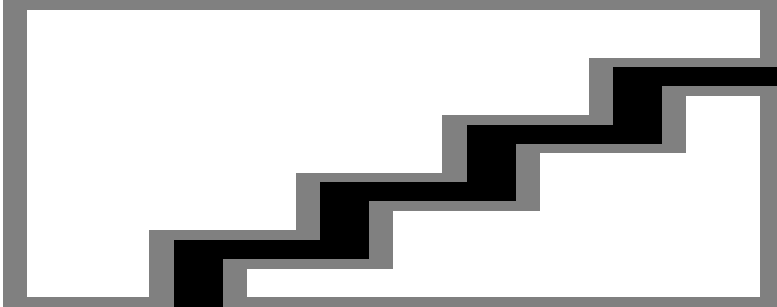
 error out contains error information. This output provides [standard error out](#) functionality.


Analog Runt Ref Trigger

Configures common properties for runt triggering.


A runt trigger occurs when both the leading edge and trailing edge of a pulse cross only one of two trigger thresholds you specify and with a polarity you specify, where the polarity is relative to the threshold crossed.

 Note You can add time qualification to trigger on runt pulses that, in addition to meeting your amplitude and polarity criteria, also meet duration criteria. Add time qualification to the runt trigger with the [Runt Time Condition](#), [Runt Time Low Limit](#), and [Runt Time High Limit](#) properties. Time qualification is disabled by default.




 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

 polarity specifies the polarity of the runt pulses, relative to the runt threshold the pulses cross, that trigger the oscilloscope for runt triggering.


Refer to the [Runt Polarity](#) property for defined values.

 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.


 low threshold specifies, in volts, the lower of two thresholds that bound the vertical range to examine for runt pulses.

Refer to the [Runt Low Threshold](#) property for more information.


 high threshold specifies, in volts, the higher of two thresholds that bound the vertical range to examine for runt pulses.

Refer to the [Runt High Threshold](#) property for more information.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

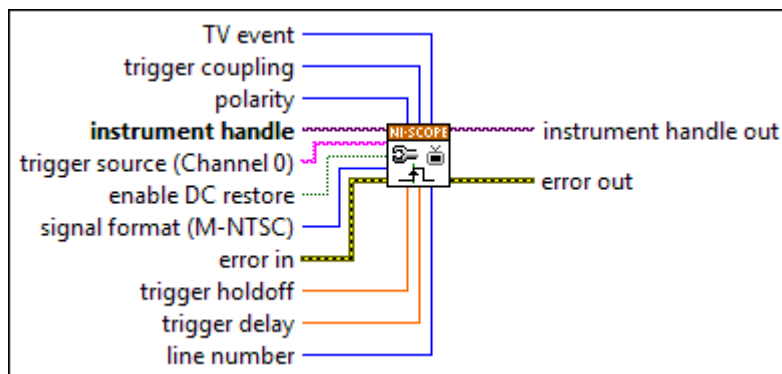
Default Value: 0.0 s

 instrument handle out has the same value as the instrument handle.


 error out contains error information. This output provides [standard error out](#) functionality.

Video Ref Trigger

Configures the common properties specific to video triggering. The video triggering properties are polarity, enable DC restore, signal format, event, and line number. A video trigger occurs when the digitizer finds a valid video signal sync. Use the trigger holdoff to skip a specific number of frames between acquisitions. For example, to acquire a specific line number multiple times and repeat the same chroma phase, skip one frame in NTSC ($121 \text{ ms} < \text{holdoff} < 159 \text{ ms}$) and five frames in SECAM ($201 \text{ ms} < \text{holdoff} < 239 \text{ ms}$).



 instrument handle identifies a particular instrument session.

 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

TF enable DC restore offsets each video line so the clamping level (the portion of the video line between the color burst and the beginning of the active image) is moved to zero volt.

Default Value: FALSE

I32 signal format specifies the video format to use.

Refer to the [Signal Format](#) property for defined values.

U16 polarity specifies the polarity of the video sync.

Default Value: negative

Defined Values:

positive

negative

U16 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

I32 TV event specifies what TV event to trigger on.

Refer to the [Event](#) property for defined values.

E+I error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.


Default Value: 0.0 s


I32 line number specifies the line number to trigger on.

The line number range covers an entire frame and is referenced as shown in [Vertical Blanking and Synchronization Signal](#).

The number of lines available depends on the format. For valid ranges, refer to the [Line Number](#) property.

Default value: 1

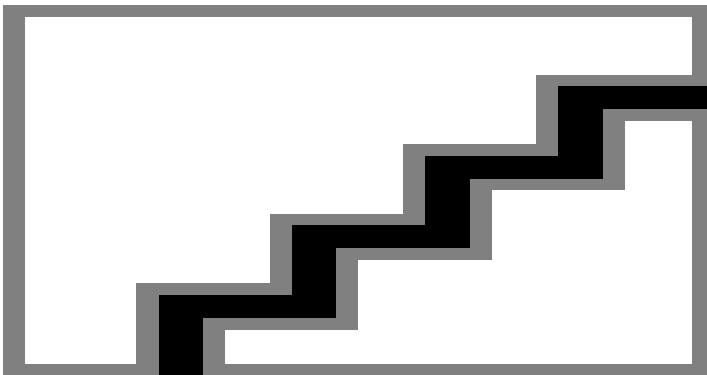
 instrument handle out has the same value as the instrument handle.


 error out contains error information. This output provides [standard error out](#) functionality.

Analog Width Ref Trigger


Configures common properties for width triggering.

A width trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a duration that is either within or outside a duration range you specify.




 low threshold specifies, in seconds, the lower bound on the range of pulse durations that triggers the oscilloscope.


Refer to the [Width Low Threshold](#) property for more information.


 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

 polarity specifies the polarity of the pulses that trigger the oscilloscope for width triggering.


Refer to the [Width Polarity](#) property for defined values.

 instrument handle identifies a particular instrument session.


 trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0


Refer to the [Trigger Source](#) property for defined values.


 trigger level specifies the voltage threshold for the trigger.

Default Value: 0.0 V


 width condition specifies whether the oscilloscope triggers on pulses of duration within or outside the range of pulse durations bounded by low threshold and high threshold.

Refer to the [Width Condition](#) property for defined values.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s


 trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

 high threshold specifies, in seconds, the upper bound on the range of pulse durations that triggers the oscilloscope.

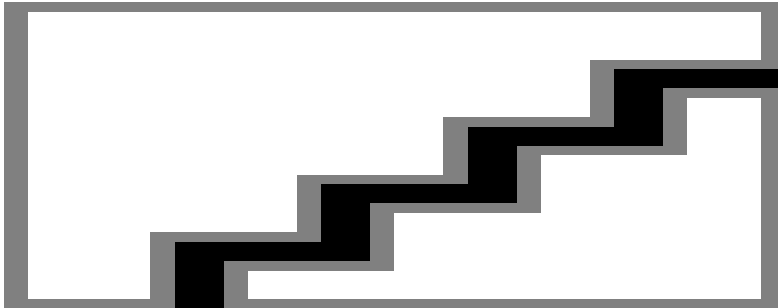
Refer to the [Width High Threshold](#) property for more information.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Analog Window Ref Trigger

Configures common properties for window triggering. A window trigger occurs when a signal enters or leaves a window you specify with the high level or low level parameters.



I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

DBL low level passes the voltage threshold for the digitizer to use for low triggering.

Default Value: 0 V

DBL high level passes the voltage threshold for the digitizer to use for high triggering.

Default Value: 0.10 V

I32 trigger window mode specifies whether the trigger should occur when the signal is entering or leaving a window.

Default Value: Entering

U16 trigger coupling specifies how to couple the trigger signal.

Refer to the [Trigger Coupling](#) property for defined values.

F+H error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

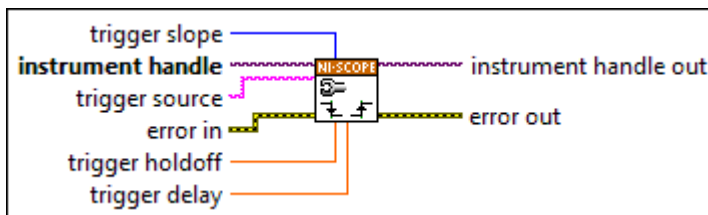
I/O instrument handle out has the same value as the instrument handle.

ERR error out contains error information. This output provides [standard error out](#) functionality.

Digital Edge Ref Trigger

Configures the common properties of a digital trigger.

Note Digital triggering is not supported in [RIS mode](#).



I/O instrument handle identifies a particular instrument session.

abc trigger source passes the source you want the digitizer to monitor for a trigger.

Default Value: Channel 0

Refer to the [Trigger Source](#) property for defined values.

I32 trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to the [Trigger Slope](#) property for defined values.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

I/O instrument handle out has the same value as the instrument handle.

FE error out contains error information. This output provides [standard error out](#) functionality.

Immediate Ref Trigger

Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.

Related topics:

- [Immediate Triggers](#)



I/O instrument handle identifies a particular instrument session.

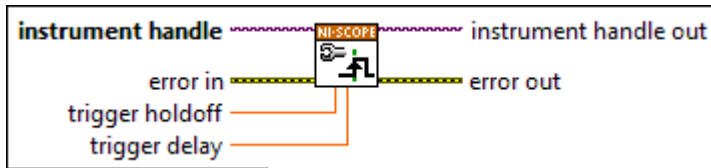
FE error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

FE error out contains error information. This output provides [standard error out](#) functionality.

Software Ref Trigger

Configures common properties for software triggering. This VI only works in an Initiate/Fetch operation.



I/O instrument handle identifies a particular instrument session.

E error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

DBL trigger holdoff specifies the length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

Default Value: 0.0 s

DBL trigger delay specifies how long the digitizer waits after it receives the trigger to start acquiring data.

Default Value: 0.0 s

E error out contains error information. This output provides [standard error out](#) functionality.

niScope Export Signal

Configures the digitizer to generate a signal that other devices can detect when configured for digital triggering or sharing clocks. The signal parameter specifies what condition causes the digitizer to generate the signal. The output terminal parameter specifies where to send the signal on the hardware (such as a PFI connector or RTSI line).

In cases where multiple instances of a particular signal exist, use the signal identifier input to specify which instance to control. For normal events, only one instance exists and you should leave signal identifier set to None. You can call this VI multiple times, and set each line available to a different signal.

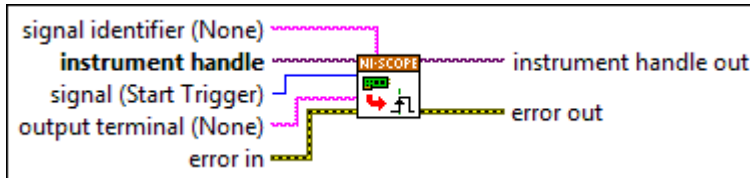
To unprogram a specific line on a device, call this VI with the signal you no longer want to export and set output terminal to None.



Note This VI replaces [niScope Configure Trigger Output](#).

Related topics:

- [Triggering](#)
- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [PXI Trigger Lines](#)



I/O instrument handle identifies a particular instrument session.

U16 signal specifies the signal (clock, trigger, or event) to export.

Defined Values

None (0)—Do not generate a digital pulse.

Reference Trigger (1)—Generate a pulse when detecting the stop/reference trigger.

Start Trigger (2)—Generate a pulse at the start of the acquisition.

End of Acquisition (3)—Generate a pulse at the end of the acquisition.

End of Record (4)—Generate a pulse at the end of each record.

Record Advance (5)—Generate a pulse when the digitizer is advancing to the next record.

Ready for Record Advance (6)—Asserts when the digitizer is ready to advance to the next record.

Ready for Start (7)—Asserts when the digitizer is initiated and ready to accept a start trigger to begin sampling.

Ready for Reference Trigger (10)—Asserts when the digitizer is ready to accept a reference trigger.

Reference Clock (100)—Export the reference clock for the digitizer to the specified terminal.

Sample Clock (101)—Export the sample clock for the digitizer to the specified terminal.

5 Volt Power (13)—Export a 5 V power source.

 output terminal identifies the hardware signal line on which the digital pulse is generated.

Defined Values

None (default)

PXI Trigger Line 0/RTSI 0

PXI Trigger Line 1/RTSI 1

PXI Trigger Line 2/RTSI 2

PXI Trigger Line 3/RTSI 3

PXI Trigger Line 4/RTSI 4

PXI Trigger Line 5/RTSI 5

PXI Trigger Line 6/RTSI 6

PXI Trigger Line 7/RTSI 7 (RTSI Clock)

PXI Star Trigger

PFI 0

PFI 1

PFI 2

PFI 3


PFI 4


PFI 5


PFI 6


PFI 7

Clock Out

 signal identifier describes the signal being exported.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Send Software Trigger Edge

Sends the selected trigger to the digitizer. If you called niScope Configure Trigger Software, call this VI when you want the reference trigger to occur. You can also call this VI to override a misused edge, digital, or hysteresis reference trigger. If you have configured an Acquisition Arm Source, an Arm Reference Trigger Source, or an Advance Trigger Source, call this VI when you want to send the corresponding trigger to the digitizer.

Related topics:

- [Software Triggers](#)



 instrument handle identifies a particular instrument session.

 which trigger specifies the type of trigger to send to the digitizer.


Defined Values


Start Trigger


Arm Reference Trigger

Reference Trigger

Advance Trigger

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

Acquisition

Owning Palette: [NI-SCOPE VIs](#)

Use the NI-SCOPE Acquisition VIs to control your data acquisition and to retrieve data from your device.

Palette Object	Description
niScope Read (poly)	Initiates an acquisition, waits for it to complete, and acquires data.
niScope Get Wfm Coefficients (poly)	Returns coefficients that can be used to scale or normalize binary waveform data.
niScope Commit	Commits to hardware all the parameter settings associated with the task. Use this VI if you want a parameter change to be immediately reflected in the hardware.
niScope Initiate Acquisition	Initiates a waveform acquisition.
niScope Fetch (poly)	Retrieves data that the digitizer has acquired from a previously initiated acquisition.
niScope Abort	Aborts an acquisition without changing the settings on the digitizer. Use this VI if the digitizer times out waiting for a trigger.
niScope Acquisition Status	Returns status information indicating whether an acquisition is in progress, complete, or unknown to the acquisition status output parameter.

niScope Read (poly)

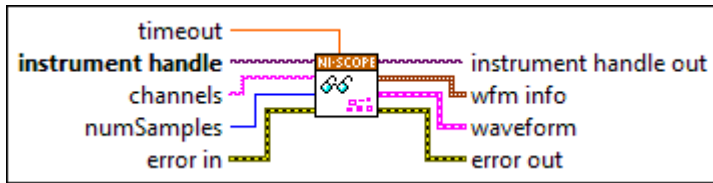
Initiates an acquisition, waits for it to complete, and acquires data.

Related topics:

- [Acquisition Functions](#)
- [NI-SCOPE Programming Flow](#)

Cluster

Initiates an acquisition, waits for it to complete, and retrieves the data for a single channel and record.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

DBL wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:


$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 reserved1 is reserved. Do not use.


 reserved2 is reserved. Do not use.

 waveform is a cluster containing the initial x value, xincrement, and a waveform array. This output can be wired directly to the LabVIEW waveform graph, so the waveform is plotted with timing information. The initial x value in the cluster is relative to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

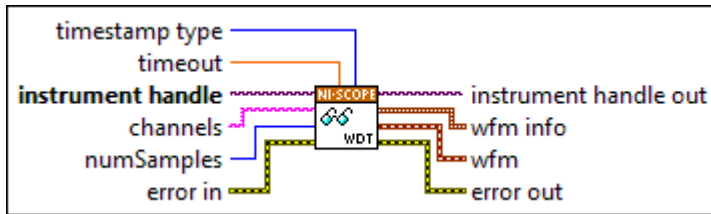
 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 wfm is an array of waveform data containing one waveform without any timing information.

 error out contains error information. This output provides [standard error out](#) functionality.

WDT

Returns the waveform the digitizer acquires for the specified channel. The VI initiates an acquisition that returns a scaled voltage waveform in a waveform data type that includes timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

I32 timestamp Type specifies the time basis for the timestamp on the WDT data.

Defined Values

absolute

relative

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

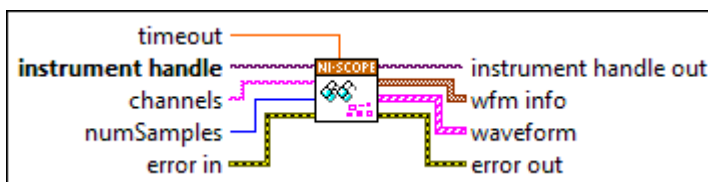
DBL reserved2 is reserved. Do not use.

wfm returns data for wiring to a graph, including the timing information.


error out contains error information. This output provides [standard error out](#) functionality.


1D Cluster


Initiates an acquisition, waits for it to complete, and retrieves the data for multiple channels and records.





I/O instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.


Use for scaling binary data with the following formula:


$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 waveform is an array of clusters, each containing the initial x value, x increment, and a waveform array. This output can be wired directly to the LabVIEW waveform graph, so each waveform in the array is plotted with timing information. The initial x value in the cluster is the value relative to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.


The array of clusters includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal there is one waveform per channel per record. If you call the fetch VI during a normal acquisition with the channel string "0,1" the order of the output is:


record 0, channel 0


record 0, channel 1


record 1, channel 0

record 1, channel 1

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

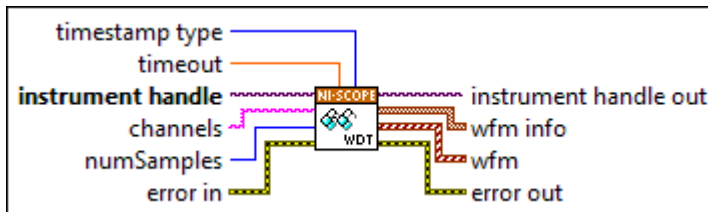
 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 wfm is an array of waveform data containing one waveform without any timing information.

 error out contains error information. This output provides [standard error out](#) functionality.

1D WDT

Initiates an acquisition and returns a one-dimensional array of LabVIEW waveform data types that includes timing information. This VI is only supported in LabVIEW 7.0 or later.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

I32 timestamp Type specifies the time basis for the timestamp on the WDT data.

Defined Values


absolute


relative


Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


I/O instrument handle out has the same value as the instrument handle.


I32 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 wfm returns a one-dimensional array of LabVIEW waveform data types that contain timing information. If you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.


For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

index 2 = record 1, channel 0

index 3 = record 1, channel 1

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Get Wfm Coefficients (poly)

Returns coefficients that can be used to scale or normalize binary waveform data.


Get Normalization Coefficients


Returns coefficients that can be used to convert binary data to normalized and calibrated data.


Refer to [Scaling and Normalization of Binary Data](#) for more information.




 instrument handle identifies a particular instrument session.

 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 coefficient info is an array of structures containing gain and offset coefficients for a given channel.

 offset is the offset factor of the given channel.

Use for normalizing binary data with the following formula:


$$\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.

Use for normalizing binary data with the following formula:

$$\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

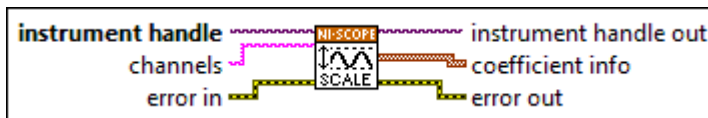
 reserved2 is reserved. Do not use.


 error out contains error information. This output provides [standard error out](#) functionality.


Get Scaling Coefficients


Returns coefficients that can be used to scale binary data to volts.


Refer to [Scaling and Normalization of Binary Data](#) for more information.




 instrument handle identifies a particular instrument session.

 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 coefficient info is an array of structures containing gain and offset coefficients for a given channel.

 offset is the offset factor of the given channel.


Use for scaling binary data with the following formula:

$$\mathbf{voltage = binary\ data \times gain\ factor + offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage = binary\ data \times gain\ factor + offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Commit

Commits to hardware all the parameter settings associated with the task. Use this VI if you want a parameter change to be immediately reflected in the hardware.


After using a Configuration VI to set a parameter, call this VI, which causes the driver to write the new configuration to the digitizer hardware immediately instead of waiting until the next call to [niScope Initiate Acquisition](#).

Related topics:


- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [NI-SCOPE Programming Flow](#)



 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Initiate Acquisition

Initiates a waveform acquisition.

After you call this VI, the digitizer leaves the Idle state and waits for a trigger. The digitizer acquires a waveform for each channel you enable with [niScope Configure Vertical](#).

Related topics

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [NI-SCOPE Programming Flow](#)



I/O instrument handle identifies a particular instrument session.

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err error out contains error information. This output provides [standard error out](#) functionality.

niScope Fetch (poly)

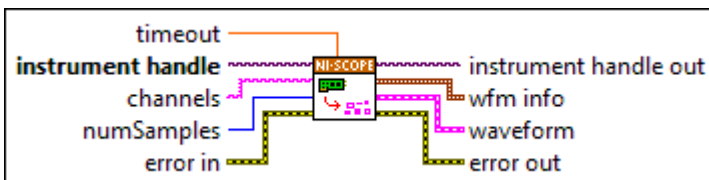
Retrieves data that the digitizer has acquired from a previously initiated acquisition.

Related topics

- [Fetching Data](#)
- [Acquiring Data Continuously](#)

Cluster

Retrieves the waveform the digitizer has acquired for the specified channel from a previously initiated acquisition. Returns scaled voltage waveforms in a cluster that includes timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

132 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter

if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

WFM wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

Cluster waveform is a cluster containing the initial x value, xIncrement, and a waveform array. This output can be wired directly to the LabVIEW waveform graph, so the waveform is plotted with timing information. The initial x value in the cluster is relative to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

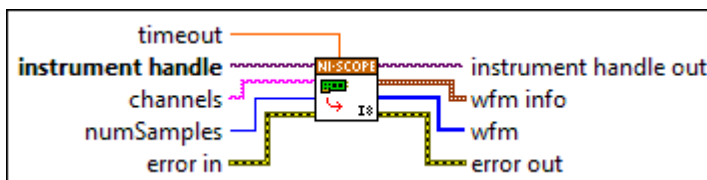
DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL wfm is an array of waveform data containing one waveform without any timing information.

Cluster error out contains error information. This output provides [standard error out](#) functionality.

1D I8

Retrieves data from a single channel and record. Returns a one-dimensional array of binary 8-bit values.





I/O instrument handle identifies a particular instrument session.


abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and


attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

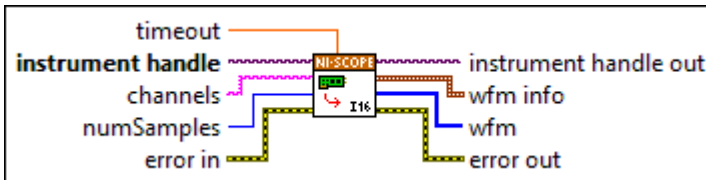
 reserved2 is reserved. Do not use.

I18 wfm returns a one-dimensional array of data.

E18 error out contains error information. This output provides [standard error out](#) functionality.

1D I16

Retrieves data from a single channel and record. Returns a one-dimensional array of binary 16-bit values.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

E18 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

E18 wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

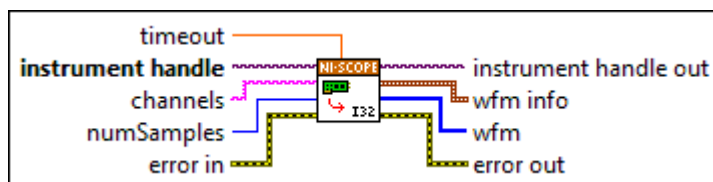
DBL reserved2 is reserved. Do not use.

I16 wfm returns a one-dimensional array of data.


ERR error out contains error information. This output provides [standard error out](#) functionality.


1D I32


Retrieves data from a single channel and record. Returns a one-dimensional array of binary 32-bit values.





I/O instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.


Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

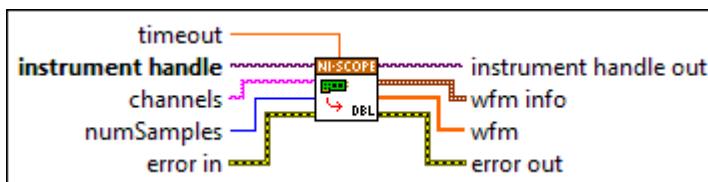
 reserved2 is reserved. Do not use.


 wfm returns a one-dimensional array of data.


 error out contains error information. This output provides [standard error out](#) functionality.


Fetch


Retrieves data from a single channel and record. Returns a one-dimensional array of scaled voltages.





 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

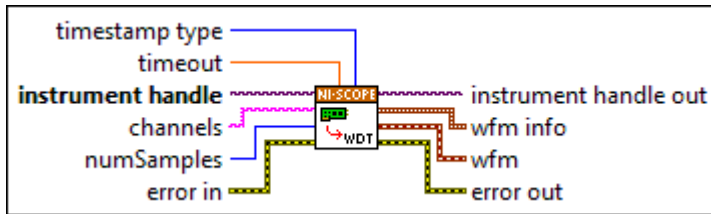
 reserved2 is reserved. Do not use.

 wfm is an array of waveform data containing one waveform without any timing information.

 error out contains error information. This output provides [standard error out](#) functionality.

WDT (Waveform Data Type)

Retrieves a waveform the digitizer has acquired for the specified channel. Returns scaled voltage data in a waveform data type that includes timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

I32 timestamp Type specifies the time basis for the timestamp on the WDT data.

Defined Values

absolute

relative

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


DBL gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

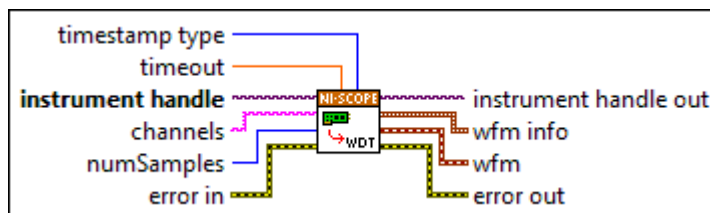
DBL reserved2 is reserved. Do not use.

 wfm returns data for wiring to a graph, including the timing information.

 error out contains error information. This output provides [standard error out](#) functionality.

Complex WDT (Waveform Data Type)

Retrieves complex waveforms the digitizer has acquired for the specified channel. Returns a two-dimensional array of LabVIEW complex waveform data types that includes timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

I32 timestamp Type specifies the time basis for the timestamp on the WDT data.

Defined Values

absolute

relative

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

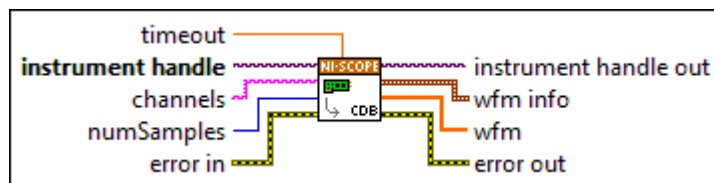
DBL reserved2 is reserved. Do not use.

Wfm wfm returns data for wiring to a graph, including the timing information.

Error error out contains error information. This output provides [standard error out](#) functionality.

1D CDB

Retrieves data from single channels and records. Returns a one-dimensional array of complex, scaled waveforms.





I/O instrument handle identifies a particular instrument session.


abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and


attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

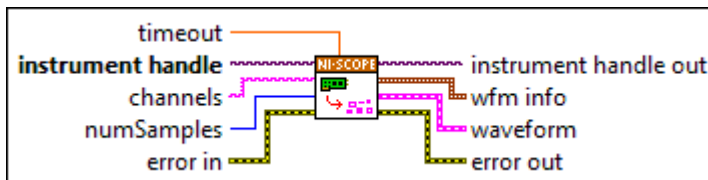
 reserved2 is reserved. Do not use.

CDB wfm is an array of waveform data containing one waveform without timing information.

ERR error out contains error information. This output provides [standard error out](#) functionality.

Cluster CDB

Retrieves data from single channels and records. Returns a complex, scaled waveform in a cluster that contains timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


I/O instrument handle out has the same value as the instrument handle.


ERR wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 `absoluteInitialX` is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


 `relativeInitialX` is the time in seconds from the trigger to the first sample in the acquired waveform.

 `xIncrement` indicates the time in seconds between two samples in the acquired waveform.

 `offset` is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 `gain` is the gain factor of the given channel.


Use for scaling binary data with the following formula:


$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 `reserved1` is reserved. Do not use.


 `reserved2` is reserved. Do not use.

 `waveform` is a cluster containing the initial time value, time increment, and a waveform array. The initial time value in the cluster is relative to the trigger. The relative initial time value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

 `t0` is the initial time value relative to the trigger, such that the trigger always occurs at time equals zero.

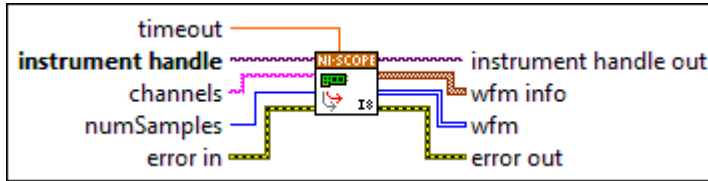
 `dt` is the time interval between two samples in the waveform. This value is 1/ (sampling rate) for time-domain acquisitions.

 `Y` is an array of waveform data containing one waveform.

 `error out` contains error information. This output provides [standard error out](#) functionality.

2D I8

Retrieves data from multiple channels and records. Returns a two-dimensional array of binary 8-bit waveforms.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


I/O instrument handle out has the same value as the instrument handle.


arr wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


 `relativeInitialX` is the time in seconds from the trigger to the first sample in the acquired waveform.

 `xIncrement` indicates the time in seconds between two samples in the acquired waveform.

 `offset` is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 `gain` is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$

 `reserved1` is reserved. Do not use.

 `reserved2` is reserved. Do not use.

 `wfm` returns a two-dimensional array of binary data; if you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.


For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

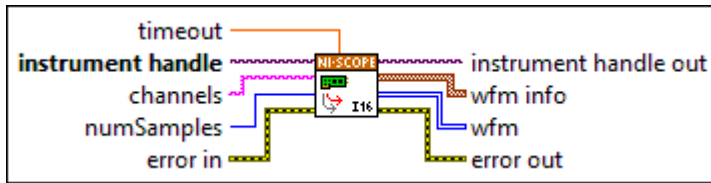
index 2 = record 1, channel 0

index 3 = record 1, channel 1

 `error out` contains error information. This output provides [standard error out](#) functionality.

2D I16

Retrieves data from multiple channels and records. Returns a two-dimensional array of binary 16-bit waveforms.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

I16 wfm returns a two-dimensional array of binary data; if you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.

For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

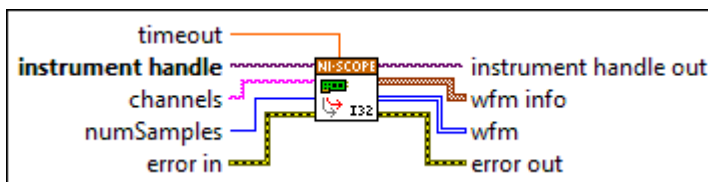
index 2 = record 1, channel 0

index 3 = record 1, channel 1


E16 error out contains error information. This output provides standard error out functionality.


2D I32


Retrieves data from multiple channels and records. Returns a two-dimensional array of binary 32-bit waveforms.





I/O instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

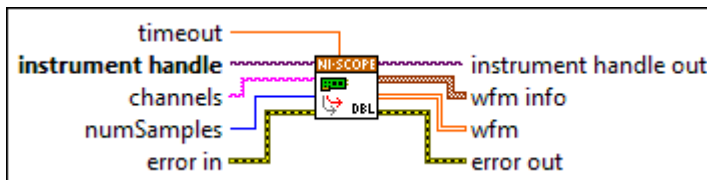
 reserved2 is reserved. Do not use.

 wfm is an array of clusters. Each waveform in the output array has some corresponding information.


 error out contains error information. This output provides [standard error out](#) functionality.


2D DBL


Retrieves data from multiple records or multiple channels. Returns a two-dimensional array of scaled voltage waveforms. This VI makes it easy for you to save data to a disk or perform math operations.





 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 wfm is an array of waveforms; that is, a two-dimensional array. This output can be wired directly to the LabVIEW waveform graph, but each waveform is plotted without timing information.

The 2D array includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal, there is one waveform per channel per

record. If you call the fetch VI during a normal acquisition with the channel string "0,1" and the record number set to -1, the order of the output is:

record 0, channel 0

record 0, channel 1

record 1, channel 0

record 1, channel 1

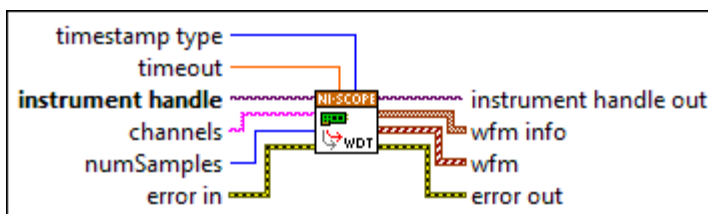
The order of the channels is the order specified by the channels parameter.



error out contains error information. This output provides [standard error out](#) functionality.

1D WDT (Waveform Data Type)

Retrieves waveforms the digitizer has acquired from multiple records or multiple channels. Returns a two-dimensional array of LabVIEW waveform data types that includes timing information.



instrument handle identifies a particular instrument session.




channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).



numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.




timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 timestamp Type specifies the time basis for the timestamp on the WDT data.


Defined Values


absolute


relative


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 wfm returns a one-dimensional array of LabVIEW waveform data types that contain timing information. If you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.


For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

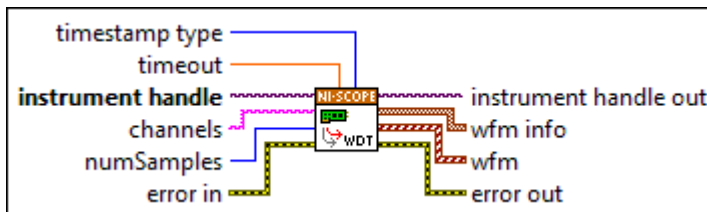
index 2 = record 1, channel 0

index 3 = record 1, channel 1


 error out contains error information. This output provides [standard error out](#) functionality.


1D Complex WDT (Waveform Data Type)

Retrieves complex waveforms the digitizer has acquired from multiple records or multiple channels. Returns a two-dimensional array of LabVIEW complex waveform data types that includes timing information.



 instrument handle identifies a particular instrument session.

 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

I32 timestamp Type specifies the time basis for the timestamp on the WDT data.

Defined Values

absolute

relative

E00 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

E00 wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

wfm returns a one-dimensional array of LabVIEW waveform data types that contain timing information. If you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.

For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

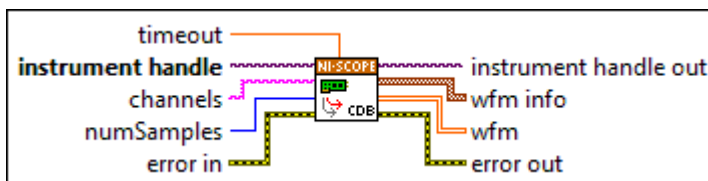
index 2 = record 1, channel 0

index 3 = record 1, channel 1

error out contains error information. This output provides [standard error out](#) functionality.

2D CDB

Retrieves data from multiple channels and records. Returns a two-dimensional array of complex, scaled waveforms.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

ERR error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

I/O instrument handle out has the same value as the instrument handle.

WFM wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.


DBL reserved2 is reserved. Do not use.

WFM wfm is an array of waveforms; that is, a two-dimensional array.

The 2D array includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal, one waveform per channel per record is acquired. If you call the fetch VI during a normal acquisition with the channel string "0,1" and the record number set to -1, the order of the output is:

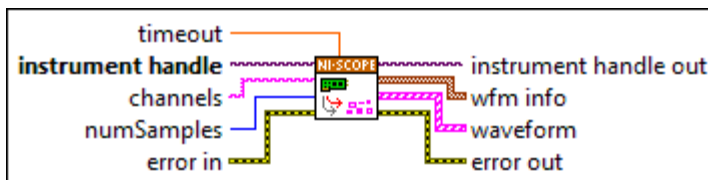
record 0, channel 0,
 record 0, channel 1,
 record 1, channel 0,
 record 1, channel 1

The order of the channels is the order specified by the channels parameter.


 error out contains error information. This output provides [standard error out](#) functionality.


1D Cluster


Retrieves the waveform the digitizer has acquired for multiple channels and records. Returns scaled voltage waveforms in a cluster that includes timing information.





 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.


 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


 gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 waveform is an array of clusters, each containing the initial x value, x increment, and a waveform array.

This output can be wired directly to the LabVIEW waveform graph, so each waveform in the array is plotted with timing information. The initial x value in the cluster is the value relative

to the trigger. The relative initial x value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

The array of clusters includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal there is one waveform per channel per record. If you call the fetch VI during a normal acquisition with the channel string "0,1" the order of the output is:

record 0, channel 0

record 0, channel 1

record 1, channel 0

record 1, channel 1

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

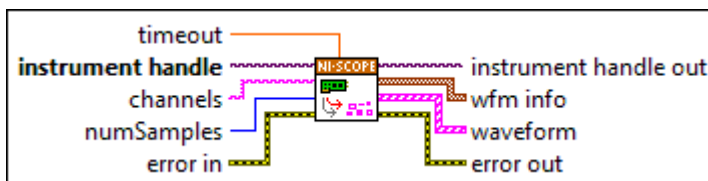
DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL wfm contains the voltage data (the y-axis data in the graph).

err error out contains error information. This output provides [standard error out](#) functionality.

Cluster CDB

Retrieves the waveform the digitizer has acquired for multiple channels and records. Returns a one-dimensional array of complex, scaled waveforms in clusters that include timing information.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 numSamples is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The VI reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the VI returns an error.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

info wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.


Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$


DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 waveform is an array of clusters, each containing the initial time value, time increment, and a waveform array. The initial time value in the cluster is the value relative to the trigger. The relative initial time value is the time from the trigger to the first point so the trigger always occurs at time equals zero.

The array of clusters includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal there is one waveform per channel per record. If you call the fetch VI during a normal acquisition with the channel string "0,1" the order of the output is:


record 0, channel 0,


record 0, channel 1,


record 1, channel 0,


record 1, channel 1,

The order of the channels is the order specified by the channels parameter.

 t0 is the initial time value relative to the trigger, such that the trigger always occurs at time equals zero.

 dt is the time interval between two samples in the waveform. This value is 1/ (sampling rate) for time-domain acquisitions.

 Y is an array of waveform data containing one waveform.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Abort

Aborts an acquisition without changing the settings on the digitizer. Use this VI if the digitizer times out waiting for a trigger.

Related topics

- [NI-SCOPE Programming Flow](#)



I/O instrument handle identifies a particular instrument session.

Error error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

Error error out contains error information. This output provides [standard error out](#) functionality.

niScope Acquisition Status

Returns status information indicating whether an acquisition is in progress, complete, or unknown to the acquisition status output parameter.



I/O instrument handle identifies a particular instrument session.

Error error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

I32 acquisition status returns whether the acquisition is in progress, complete, or unknown.

Defined Values

Acquisition in progress (0)

Acquisition complete (1)

Status unknown (-1)

Error error out contains error information. This output provides [standard error out](#) functionality.

Calibration

Owning Palette: [NI-SCOPE VIs](#)


Use the NI-SCOPE Calibration VIs to self-calibrate your digitizer and to access lower-level external calibration functions.

PaletteObject	Description
niScope Cal Self Calibrate	Self-calibrates most NI digitizers, including all SMC-based devices.
Subpalette	Description
External Calibration	Use the NI-SCOPE External Calibration VIs to externally calibrate your digitizer. Some of these functions are device-specific; refer to the manual calibration procedure for your device for more information.

niScope Cal Self Calibrate


Self-calibrates most NI digitizers, including all SMC-based devices.

For SMC-based devices, if the self-calibration is performed successfully in a regular session, the calibration constants are immediately stored in the self-calibration area of the EEPROM. If the self-calibration is performed in an external calibration session, the calibration constants take effect immediately for the duration of the session. However, they are not stored in the EEPROM until you call [niScope Cal End](#) with action set to Store Calibration and no errors occur.

 **Note** To verify that your digitizer supports self-calibration, refer to [Features Supported by Device](#).



 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).


 Option allows you to self-calibrate all channels or restore the external calibration.


Defined Values

Self-Calibrate All Channels (Default)

Restore External Calibration

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

External Calibration

Owning Palette: [Calibration](#)

Use the NI-SCOPE External Calibration VIs to externally calibrate your digitizer. Some of these functions are device-specific; refer to the manual calibration procedure for your device for more information.

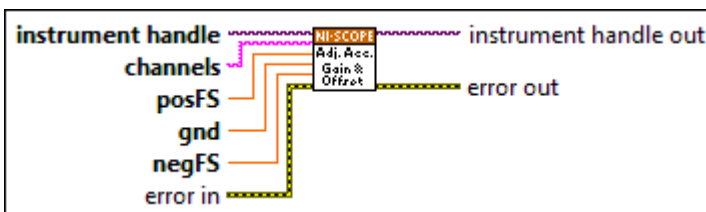
Palette Object	Description
niScope Cal Start	Opens an external calibration session and produces a calibration session handle that is required by the external calibration VIs.
niScope Cal End	Closes an external calibration session. You must call this VI each time you call niScope Cal Start , even if an error occurs during calibration.
niScope Cal Change Password	Verifies the old password against the one stored in the EEPROM.
niScope Cal Store Misc Info	Allows you to store miscellaneous information in the EEPROM. For example, you can store an operator ID for the person or company performing

	a calibration. The information is stored immediately.
<u>niScope Cal Fetch Date</u>	Returns the last self-calibration, external calibration, or manufacturer calibration date.
<u>niScope Cal Fetch Count</u>	Returns the number of times the digitizer has been calibrated.
<u>niScope Cal Fetch Temperature</u>	For SMC-based digitizers, this VI returns the onboard temperature of the digitizer at the time of the last self-calibration or external calibration, in degrees Celsius.
<u>niScope Cal Fetch Misc Info</u>	Returns the miscellaneous information you can store during an external calibration using <u>niScope Cal Store Misc Info</u> .
<u>niScope Cal Adjust Range</u>	This VI externally calibrates the vertical range for the specified channel and vertical range setting.
<u>niScope Cal Adjust Compensation Attenuator</u>	This VI externally calibrates the compensation attenuator.
<u>niScope Cal Adjust Offset Range</u>	This VI externally calibrates the vertical offset.
<u>niScope Cal Adjust VCXO</u>	For SMC-based devices, this VI calibrates the sample rate of the digitizer.
<u>niScope Cal Adjust DCM</u>	For PXIe/PXI/PCI-5105 digitizers, this VI calibrates the external clock digital clock managers (DCMs). DCM calibration ensures that data can be sampled at the correct time in the clock period.
<u>niScope Cal Adjust Frequency Response</u>	For the PXIe-5622, calibrates the frequency response of the device. The VI is called multiple times, one for each frequency point in the sweep. The user must supply the stimulus frequency in hertz and stimulus amplitude in volts of the input signal.


niScope Cal Adjust Accessory Gain And Offset	For the PXI-5900 differential amplifier, this VI calibrates the gain and offset.
niScope Cal Set Accessory Source	For the PXI-5900 differential amplifier, this VI connects the specified channel to the calibration source and sets the calibration source to +/-10 V or to GND.
niScope Cal Measure RIS Distribution	Calls niScope Read Waveform 2,000 times to take an acquisition from the specified channel and retrieve the initial X value, which includes the time-to-digital conversion.
Subpalette	Description
5110 Ext Cal	Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5110.
5111 Ext Cal	Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5111.
5113 Ext Cal	Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5113.


niScope Cal Adjust Accessory Gain And Offset


For the PXI-5900 differential amplifier, this VI calibrates the gain and offset.





I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.


 channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).


 posFS is calculated during the external calibration procedure.

 gnd is calculated during the external calibration procedure.

 negFS is calculated during the external calibration procedure.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Adjust Compensation Attenuator


This VI externally calibrates the compensation attenuator.





 instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

 channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).

 range (V) is the vertical range to calibrate.

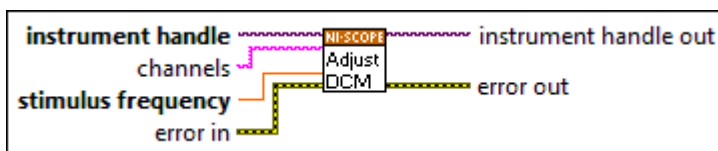
 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Adjust DCM

For PXIe/PXI/PCI-5105 digitizers, this VI calibrates the external clock digital clock managers (DCMs). DCM calibration ensures that data can be sampled at the correct time in the clock period.



I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

abc channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).

DBL stimulus frequency is the external stimulus applied to the digitizer.

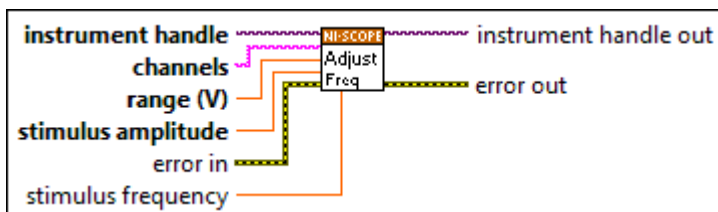
Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.









I/O instrument handle out has the same value as the instrument handle.

Err error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Adjust Frequency Response

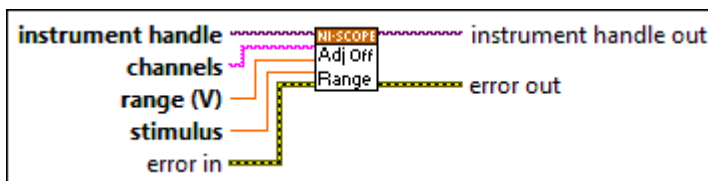
For the PXIe-5622, calibrates the frequency response of the device. The VI is called multiple times, one for each frequency point in the sweep. The user must supply the stimulus frequency in hertz and stimulus amplitude in volts of the input signal.








-  instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.
-  channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).
-  range (V) is the vertical range to calibrate.
-  stimulus amplitude is the amplitude of the external stimulus applied to the digitizer. For valid values, refer to the calibration procedure document for the device.
-  stimulus frequency is the external stimulus applied to the digitizer.
-  error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
-  instrument handle out has the same value as the instrument handle.
-  error out contains error information. This output provides [standard error out](#) functionality.


niScope Cal Adjust Offset Range


This VI externally calibrates the vertical offset.



-  instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.
-  channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).
-  range (V) is the vertical range to calibrate.
-  stimulus is the voltage of the applied DC signal.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

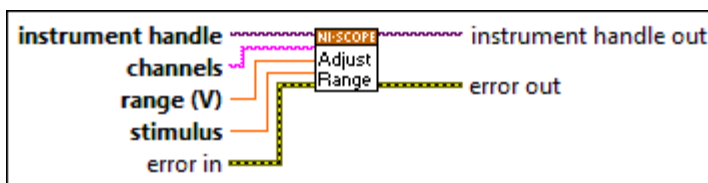
 error out contains error information. This output provides [standard error out](#) functionality.


niScope Cal Adjust Range


This VI externally calibrates the vertical range for the specified channel and vertical range setting.

Related topics:

- [Coercions of Vertical Parameters](#)




 instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.


 channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).

 range (V) is the vertical range to calibrate.

 stimulus is the voltage of the applied DC signal.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.

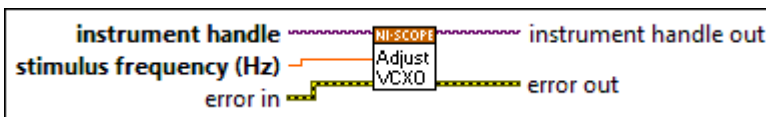
 error out contains error information. This output provides [standard error out](#) functionality.


niScope Cal Adjust VCXO


For SMC-based devices, this VI calibrates the sample rate of the digitizer.


The VI adjusts the frequency of the voltage controlled crystal oscillator (VCXO) that serves as the digitizer's onboard sample rate timebase. Check the calibration procedure specific to your device and set the value of frequency parameter. Before calling this VI, connect an accurate, stable reference signal to channel 0. (The channel used is not configurable.) The VI adjusts frequency calibration constants until the digitizer measures the frequency of the reference signal. The new calibration constants take effect immediately for the duration of the external calibration session. The constants are written to EEPROM if you call [niScope Cal End](#) with no errors and with action set to Store Calibration.


 Note Use this VI only when following the external calibration procedure for your device (Start»All Programs»National Instruments»NI-SCOPE»Documentation»Calibration).




 instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

 stimulus frequency (Hz) is the frequency of the external reference clock connected to channel 0.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

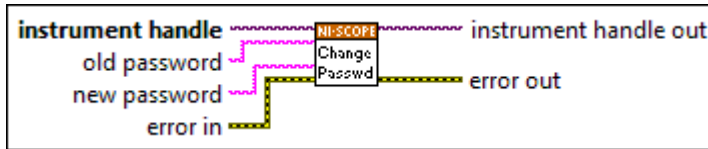
 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Change Password

Verifies the old password against the one stored in the EEPROM.

If the two match, the VI stores the new password in the EEPROM. The password is stored as four characters, but shorter strings are acceptable. For most digitizers, the default password is an empty string. For SMC-based devices, the default password is "NI". If you forget your password, call National Instruments for assistance.



I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

abc old password is the password currently stored in the EEPROM.

abc new password is the new password to store in the EEPROM. A maximum of 4 characters can be stored.

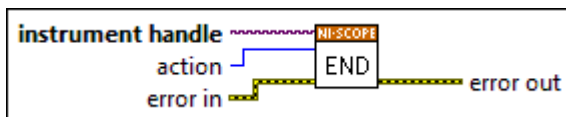
err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

err error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal End


Closes an external calibration session. You must call this VI each time you call [niScope Cal Start](#), even if an error occurs during calibration.




I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

I32 action either stores the calibration constants or aborts the calibration and discards any calibration results.

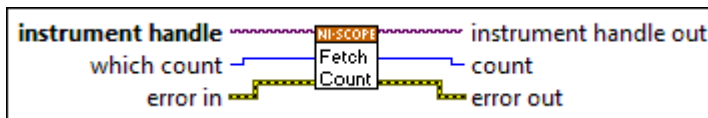
Store Calibration (0)	Stores the new calibration constants in the EEPROM. For most oscilloscopes, the current system date and the incremented calibration count are also stored. For SMC-based oscilloscopes, the current system date and on-board temperature are also stored.
Abort Calibration (1)	Closes the session and discards any new calibration constants. Some devices may write to the EEPROM during calibration, in which case this action restores the EEPROM to its original state.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 error out contains error information. This output provides [standard error out](#) functionality.


niScope Cal Fetch Count


Returns the number of times the digitizer has been calibrated.





 instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

 which count specifies which calibration count to return (self-calibration or external calibration).

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

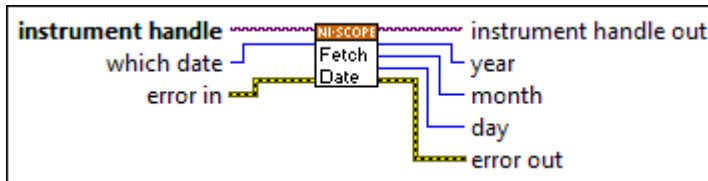
 instrument handle out has the same value as the instrument handle.









 count is the number of times the digitizer has been calibrated.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Fetch Date

Returns the last self-calibration, external calibration, or manufacturer calibration date.



-  instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.
-  which date specifies which calibration count to return (self-calibration, external calibration, or manufacturer calibration).
-  error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
-  instrument handle out has the same value as the instrument handle.
-  year is the year of the last calibration, such as 2007.
-  month is the month of the last calibration (1–12).
-  day is the day of the last calibration (1–31).
-  error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Fetch Misc Info

Returns the miscellaneous information you can store during an external calibration using [niScope Cal Store Misc Info](#).



I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

E+H error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

abc miscellaneous information is four characters that are stored in the EEPROM; however, it can be fewer than four characters if NULL-terminated.

E+H error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Fetch Temperature

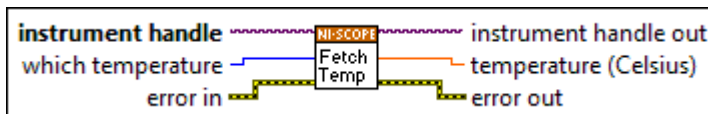
For SMC-based digitizers, this VI returns the onboard temperature of the digitizer at the time of the last self-calibration or external calibration, in degrees Celsius.

The temperature returned by this VI is an onboard temperature read from a sensor on the surface of the digitizer. This temperature should not be confused with the environmental temperature of the digitizer's surroundings. During operation, the onboard temperature is normally higher than the environmental temperature.

Temperature-sensitive parameters are calibrated during self-calibration. Therefore, the self-calibration temperature is usually the more important one to read.


Related topics


- [Thermal Shutdown](#)
- [PXI/PXIe Chassis Cooling](#)





I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

I32 which temperature specifies the calibration temperature to return, either the self-calibration or external calibration temperature.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 temperature (Celsius) is the returned temperature of the last successful calibration in degrees Celsius.

 error out contains error information. This output provides [standard error out](#) functionality.

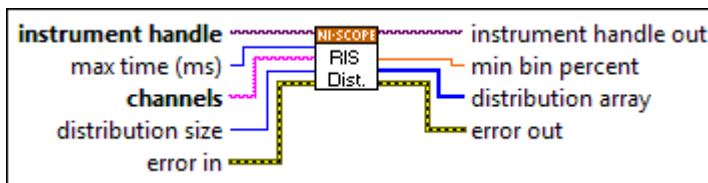
niScope Cal Measure RIS Distribution


Calls niScope Read Waveform 2,000 times to take an acquisition from the specified channel and retrieve the initial X value, which includes the time-to-digital conversion.


The time-to-digital conversion should be a uniform distribution between two sample points, because triggers should occur randomly. To test this distribution, the distribution of initial X values is created. The percentage of triggers in the smallest bin of this distribution is returned for comparison to a specification to determine if RIS is operating correctly.


Related topics

- [Random Interleaved Sampling \(RIS\)](#)



 instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

 max time (ms) is the maximum time to allow for each acquisition.

 channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).

I32 distribution size is the number of bins in the initial x distribution.

E+I error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

DBL min bin percent is the percent of triggers in the minimum bin (0–100.0).

I32 distribution array is the returned distribution of trigger times.

E+I error out contains error information. This output provides [standard error out](#) functionality.

niScope Cal Set Accessory Source

For the PXI-5900 differential amplifier, this VI connects the specified channel to the calibration source and sets the calibration source to +/-10 V or to GND.



I/O instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.

abc channels specifies the channel(s) to calibrate. For more information, refer to [Channel String Syntax](#).

I32 calSource specifies calibration signal source.

Default Value: GND

Defined Values

GND

-10V

+10V

E+I error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.


niScope Cal Start

Opens an external calibration session and produces a calibration session handle that is required by the external calibration VIs.

All other calibration VIs, such as verification and fetch VIs, work with both a calibration session and a session handle obtained from niScope Initialize. Acceptable session handles are documented for each VI in the manual calibration procedure document for each device.

Only one session handle can be obtained at a time, and every session started with this VI must be closed by calling [niScope Cal End](#). If you fail to close the session, you must unload the `niScope_32.dll` by closing your application or application development environment (ADE) before you can open another session.



 resource name specifies the device name assigned by Measurement & Automation Explorer (MAX).

Examples


Example	Device Type	Syntax
1	NI-DAQmx device	myDAQmxDevice (myDAQmx Device = device name)
2	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::2 (2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)


For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You

can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.


An alternative syntax for NI-DAQmx devices consists of DAQ::NI-DAQmx device name, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.


You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.


 **Note** NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

 password is the password that must match the password in the EEPROM to produce a valid calibration session.

The default password for most digitizers is the empty string, "". The default password for SMC-based devices is "NI".

 error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

 instrument handle identifies a particular instrument session.






 error out contains error information. This output provides standard error out functionality.

niScope Cal Store Misc Info

Allows you to store miscellaneous information in the EEPROM. For example, you can store an operator ID for the person or company performing a calibration. The information is stored immediately.

Four characters are stored in the EEPROM, and nonprintable characters are valid. However, NULL is treated as an end-of-string marker, and all characters following the first NULL are set to NULL.



-  instrument handle is the instrument handle that you obtain from [niScope Cal Start](#). The handle identifies a particular instrument session.
-  miscellaneous information is four characters that are stored in the EEPROM; however, can be fewer than four if NULL-terminated.
-  error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
-  instrument handle out has the same value as the instrument handle.
-  error out contains error information. This output provides [standard error out](#) functionality.

5110 Ext Cal

Owning Palette: [External Calibration](#)

Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5110.

Palette Object	Description
Open Ext Cal Session	Opens an external calibration session for the specified device.
DC Reference Cal Initialize	Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.
DC Reference Cal Configure	Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.
DC Reference Cal Adjust	Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.

Compensated Attenuator Cal Initialize	Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.
Compensated Attenuator Cal Configure	Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.
Compensated Attenuator Cal Adjust	Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.
Get Misc Info	Returns the text information set by the Set Misc Info VI of this library.
Set Misc Info	Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.
Get Adjustment Date and Temperature	Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.
Close Ext Cal Session	Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.
Subpalette	Description
Utilities	External calibration utilities for the PXIe-5110.

Open Ext Cal Session

Open Ext Cal Session

Opens an external calibration session for the specified device.


This VI returns an error if you specify a device that is not supported by this calibration library. This VI also downloads the calibration bitstream to the device FPGA if not already present.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name identifies the device to perform an external calibration on.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session returns a new external calibration session reference for the device specified by resource name. Use this reference as an input to other external calibration VIs in this library.

 error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Initialize

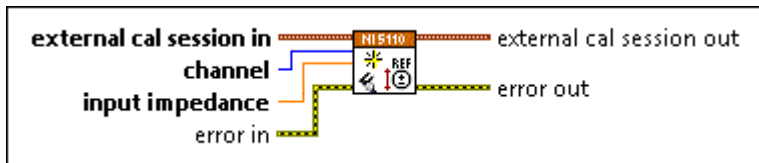
DC Reference Cal Initialize


Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 channel specifies the input channel on which to perform the DC reference adjustment.

 input impedance specifies the input impedance to apply for the DC reference adjustment.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Configure

DC Reference Cal Configure

Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111

- PXIe-5113



external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.

error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

external cal session out passes a reference to your session to the next VI.

voltage to generate (V) specifies the DC voltage to generate for the current test point.

error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Adjust

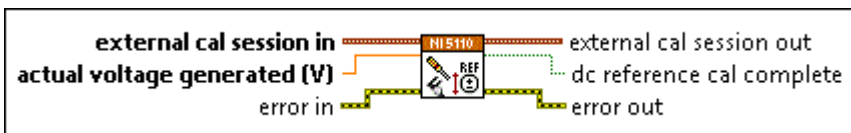
DC Reference Cal Adjust


Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.


 actual voltage generated (V) specifies the voltage applied to the channel under adjustment.

 **Note** This voltage can vary by as much as $\pm 5\%$ from the requested voltage as defined by the voltage to generate parameter of [DC Reference Cal Configure](#). If the variation exceeds $\pm 5\%$, NI-SCOPE generates an error.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 dc reference cal complete indicates whether the DC reference calibration is complete for the specified channel and the specified impedance.

 error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Initialize

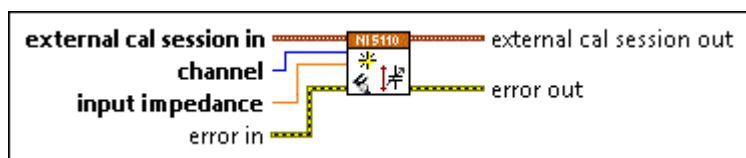
Compensated Attenuator Cal Initialize


Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.

 channel specifies the input channel on which to perform the compensated attenuator adjustment.

 input impedance specifies the input impedance to apply for the compensated attenuator adjustment.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Configure

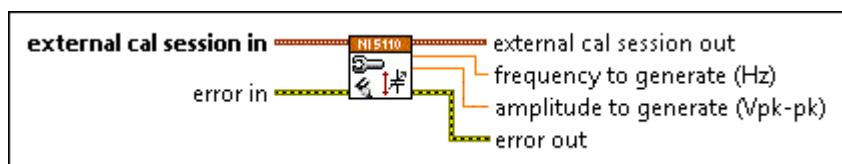
Compensated Attenuator Cal Configure


Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 frequency to generate (Hz) specifies the frequency of the square waveform to generate for the current test point.

 amplitude to generate (Vpk-pk) specifies the amplitude of the square waveform to generate for the current test point.

 error out contains error information. This output provides [standard error out](#) functionality.

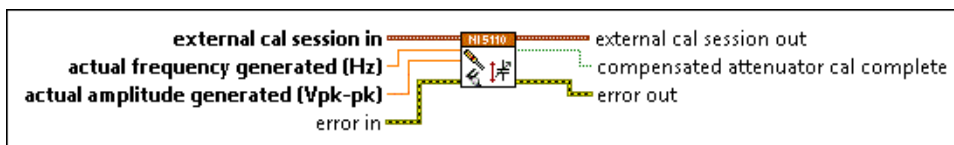
Compensated Attenuator Cal Adjust Compensated Attenuator Cal Adjust


Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.


 actual frequency generated (Hz) specifies the frequency of the square waveform applied to the channel under adjustment.


 Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the frequency to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.


 actual amplitude generated (Vpk-pk) specifies the amplitude of the square waveform applied to the channel under adjustment.

 Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the amplitude to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 compensated attenuator cal complete indicates whether the compensated attenuator calibration is complete for the specified channel and the specified impedance.

 error out contains error information. This output provides [standard error out](#) functionality.

Get Misc Info

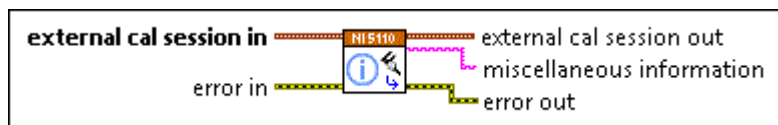
Get Misc Info


Returns the text information set by the [Set Misc Info](#) VI of this library.


If you have not yet run the [Set Misc Info](#) VI during the calibration session, this VI will return the miscellaneous information that was last committed to the device.


Supported Devices:

- PXle-5110
- PXle-5111
- PXle-5113




 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 miscellaneous information returns the text information.

 error out contains error information. This output provides [standard error out](#) functionality.

Set Misc Info

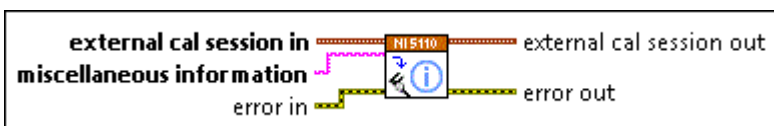
Set Misc Info


Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.


Committing this information to the device allows you to access it in subsequent calibration sessions with the [Get Misc Info](#) VI of this library.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 miscellaneous information is the text information to store.

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **external cal session out** passes a reference to your session to the next VI.

 **error out** contains error information. This output provides [standard error out](#) functionality.

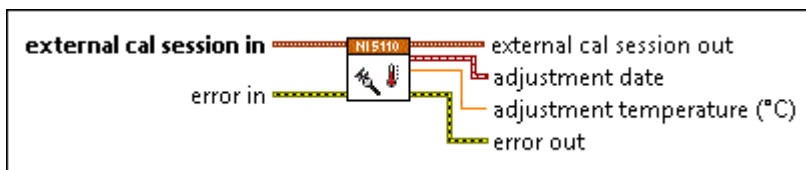
Get Adjustment Date And Temperature


Get Adjustment Date and Temperature


Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 **external cal session in** identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **external cal session out** passes a reference to your session to the next VI.

 **adjustment date** returns the date and time of the last successful external calibration.

 **adjustment temperature (° C)** returns the temperature, in degrees Celsius, of the last successful external calibration.

 **error out** contains error information. This output provides [standard error out](#) functionality.

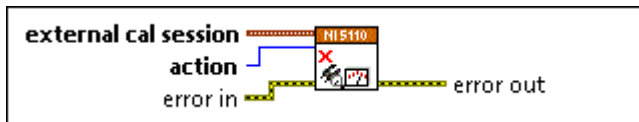
Close Ext Cal Session

Close Ext Cal Session


Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 **external cal session** is the external calibration session to close.

 **action** specifies the action to take if the external adjustment procedure completed without any errors or if the errors were corrected and the adjustment step was rerun using the same external calibration session.

Cancel (0)	Disregard the new calibration constants without changing any of the calibration data stored in the nonvolatile memory of the oscilloscope.
Commit (1)	Store the new calibration constants, adjustment time, adjustment date, and adjustment temperature in the nonvolatile memory of the oscilloscope.

Default Value: Cancel

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **error out** contains error information. This output provides [standard error out](#) functionality.

Utilities

Utilities

Owning Palette: [5110 Ext Cal](#)
 External calibration utilities for the PXIe-5110.

PaletteObject	Description
Change External Cal Password	Changes the password used to begin an external calibration session.
Get External Cal Due Date	Returns the current external calibration due date of the device.
Set External Cal Due Date	Sets the external calibration due date of the device.
Get External Cal Verification Date	Returns the most recent external calibration verification date of the device.
Set External Cal Verification Date	Sets the most recent external calibration verification date of the device.

Change External Cal Password


Change External Cal Password


Changes the password used to begin an external calibration session.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 new external cal password is the new password to store in the EEPROM of the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Get External Cal Due Date

Get External Cal Due Date


Returns the current external calibration due date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal due date returns the date and time of the external calibration due date.

 error out contains error information. This output provides [standard error out](#) functionality.

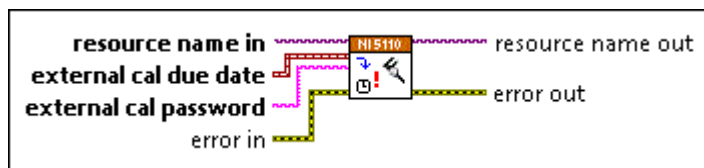
Set External Cal Due Date


Set External Cal Due Date


Sets the external calibration due date of the device.


Supported Devices:

- PXle-5110
- PXle-5111
- PXle-5113





 resource name in identifies the device.


 external cal due date sets the new due date and time of the external calibration.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

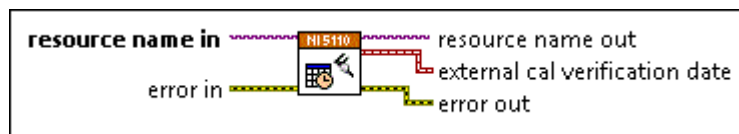
Get External Cal Verification Date


Get External Cal Verification Date


Returns the most recent external calibration verification date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal verification date returns the date and time of the most recent successful external verification of the device.

 error out contains error information. This output provides [standard error out](#) functionality.

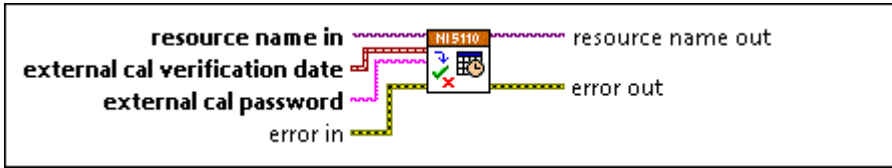
Set External Cal Verification Date

Set External Cal Verification Date


Sets the most recent external calibration verification date of the device.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal verification date sets the updated date and time of the most recent successful external verification of the device.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

5111 Ext Cal

Owning Palette: [External Calibration](#)

Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5111.

Palette Object	Description
Open Ext Cal Session	Opens an external calibration session for the specified device.
DC Reference Cal Initialize	Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.
DC Reference Cal Configure	Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.

DC Reference Cal Adjust	Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.
Compensated Attenuator Cal Initialize	Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.
Compensated Attenuator Cal Configure	Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.
Compensated Attenuator Cal Adjust	Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.
Get Misc Info	Returns the text information set by the Set Misc Info VI of this library.
Set Misc Info	Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.
Get Adjustment Date and Temperature	Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.
Close Ext Cal Session	Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.
Subpalette	Description
Utilities	External calibration utilities for the PXIe-5111.

Open Ext Cal Session

Open Ext Cal Session

Opens an external calibration session for the specified device.


This VI returns an error if you specify a device that is not supported by this calibration library. This VI also downloads the calibration bitstream to the device FPGA if not already present.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 **resource name** identifies the device to perform an external calibration on.

 **external cal password** is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **external cal session** returns a new external calibration session reference for the device specified by resource name. Use this reference as an input to other external calibration VIs in this library.

 **error out** contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Initialize

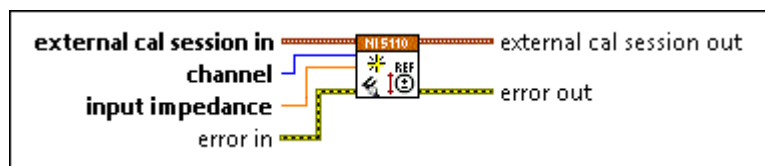
DC Reference Cal Initialize

Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

channel specifies the input channel on which to perform the DC reference adjustment.

input impedance specifies the input impedance to apply for the DC reference adjustment.

error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

external cal session out passes a reference to your session to the next VI.

error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Configure

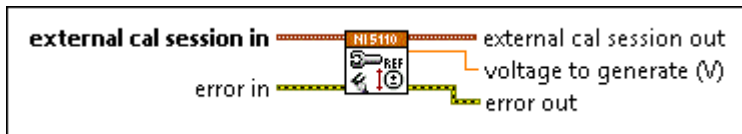
DC Reference Cal Configure

Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



EDG external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

EDG external cal session out passes a reference to your session to the next VI.

DBL voltage to generate (V) specifies the DC voltage to generate for the current test point.

ERR error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Adjust

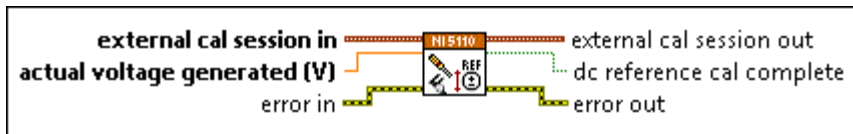
DC Reference Cal Adjust


Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.


 actual voltage generated (V) specifies the voltage applied to the channel under adjustment.

 **Note** This voltage can vary by as much as $\pm 5\%$ from the requested voltage as defined by the voltage to generate parameter of [DC Reference Cal Configure](#). If the variation exceeds $\pm 5\%$, NI-SCOPE generates an error.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 dc reference cal complete indicates whether the DC reference calibration is complete for the specified channel and the specified impedance.

 error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Initialize

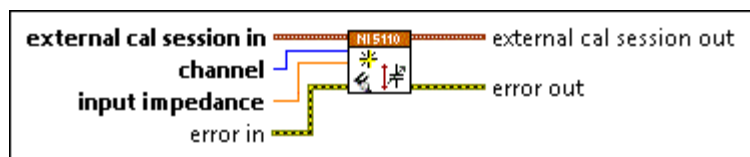
Compensated Attenuator Cal Initialize


Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 channel specifies the input channel on which to perform the compensated attenuator adjustment.

 input impedance specifies the input impedance to apply for the compensated attenuator adjustment.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Configure

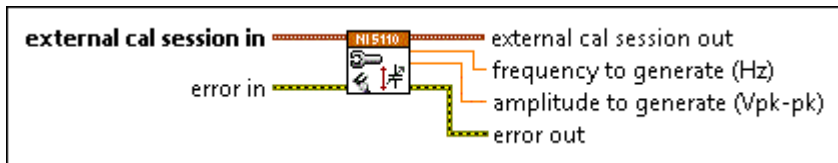
Compensated Attenuator Cal Configure

Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

external cal session out passes a reference to your session to the next VI.

frequency to generate (Hz) specifies the frequency of the square waveform to generate for the current test point.

amplitude to generate (Vpk-pk) specifies the amplitude of the square waveform to generate for the current test point.

error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Adjust

Compensated Attenuator Cal Adjust

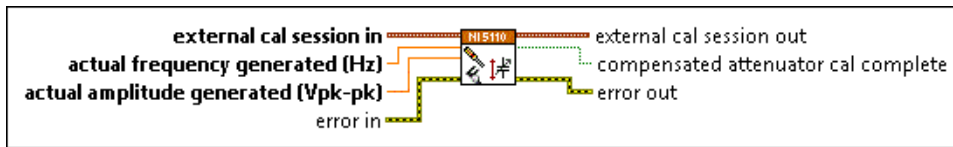
Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110

- PXIe-5111
- PXIe-5113



external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

actual frequency generated (Hz) specifies the frequency of the square waveform applied to the channel under adjustment.

Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the frequency to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.

actual amplitude generated (Vpk-pk) specifies the amplitude of the square waveform applied to the channel under adjustment.

Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the amplitude to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.

error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

external cal session out passes a reference to your session to the next VI.

compensated attenuator cal complete indicates whether the compensated attenuator calibration is complete for the specified channel and the specified impedance.

error out contains error information. This output provides [standard error out](#) functionality.

Get Misc Info

Get Misc Info


Returns the text information set by the [Set Misc Info](#) VI of this library.


If you have not yet run the [Set Misc Info](#) VI during the calibration session, this VI will return the miscellaneous information that was last committed to the device.


Supported Devices:

- PXle-5110
- PXle-5111
- PXle-5113




 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 miscellaneous information returns the text information.

 error out contains error information. This output provides [standard error out](#) functionality.

Set Misc Info

Set Misc Info

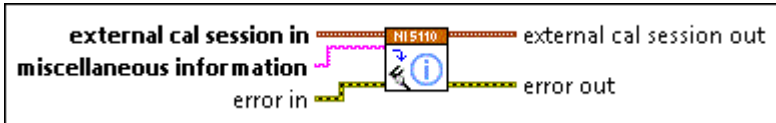
Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.


Committing this information to the device allows you to access it in subsequent calibration sessions with the [Get Misc Info](#) VI of this library.


Supported Devices:


- PXle-5110
- PXle-5111


- PXIe-5113




 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.

 miscellaneous information is the text information to store.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

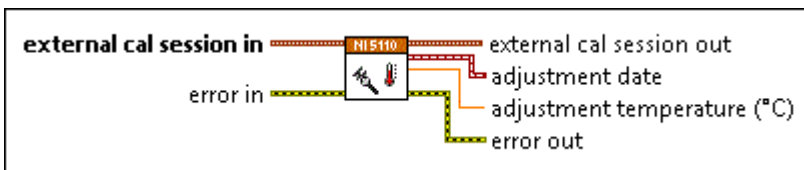
Get Adjustment Date And Temperature


Get Adjustment Date and Temperature


Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 adjustment date returns the date and time of the last successful external calibration.

 adjustment temperature (° C) returns the temperature, in degrees Celsius, of the last successful external calibration.

 error out contains error information. This output provides [standard error out](#) functionality.

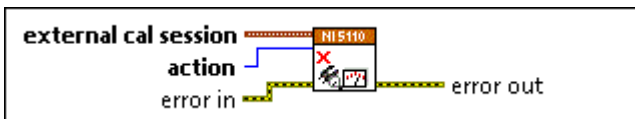
Close Ext Cal Session

Close Ext Cal Session


Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



 external cal session is the external calibration session to close.


 action specifies the action to take if the external adjustment procedure completed without any errors or if the errors were corrected and the adjustment step was rerun using the same external calibration session.


Cancel (0)

Disregard the new calibration constants without changing any of the calibration data stored in the nonvolatile memory of the oscilloscope.

Commit (1)	Store the new calibration constants, adjustment time, adjustment date, and adjustment temperature in the nonvolatile memory of the oscilloscope.
------------	--

Default Value: Cancel

 error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

 error out contains error information. This output provides standard error out functionality.

Utilities

Utilities

Owning Palette: [5111 Ext Cal](#)

External calibration utilities for the PXIe-5111.

PaletteObject	Description
Change External Cal Password	Changes the password used to begin an external calibration session.
Get External Cal Due Date	Returns the current external calibration due date of the device.
Set External Cal Due Date	Sets the external calibration due date of the device.
Get External Cal Verification Date	Returns the most recent external calibration verification date of the device.
Set External Cal Verification Date	Sets the most recent external calibration verification date of the device.

Change External Cal Password

Change External Cal Password


Changes the password used to begin an external calibration session.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 new external cal password is the new password to store in the EEPROM of the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Get External Cal Due Date


Get External Cal Due Date


Returns the current external calibration due date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal due date returns the date and time of the external calibration due date.

 error out contains error information. This output provides [standard error out](#) functionality.

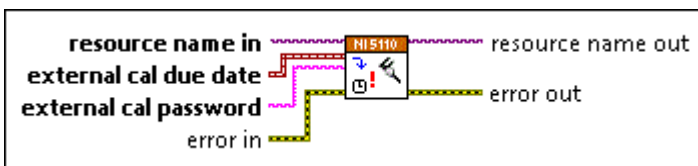
Set External Cal Due Date

Set External Cal Due Date


Sets the external calibration due date of the device.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal due date sets the new due date and time of the external calibration.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

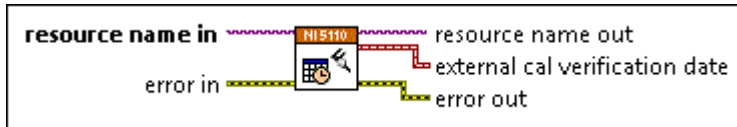
Get External Cal Verification Date

Get External Cal Verification Date


Returns the most recent external calibration verification date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal verification date returns the date and time of the most recent successful external verification of the device.

 error out contains error information. This output provides [standard error out](#) functionality.

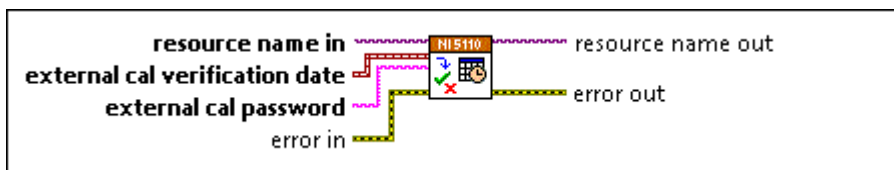
Set External Cal Verification Date

Set External Cal Verification Date


Sets the most recent external calibration verification date of the device.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal verification date sets the updated date and time of the most recent successful external verification of the device.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

5113 Ext Cal

Owning Palette: [External Calibration](#)

Use the NI-SCOPE 5110 Ext Cal VIs to perform external calibration of the DC reference and compensated attenuator on the PXIe-5113.

Palette Object	Description
<u>Open Ext Cal Session</u>	Opens an external calibration session for the specified device.
<u>DC Reference Cal Initialize</u>	Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.
<u>DC Reference Cal Configure</u>	Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.
<u>DC Reference Cal Adjust</u>	Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.
<u>Compensated Attenuator Cal Initialize</u>	Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.
<u>Compensated Attenuator Cal Configure</u>	Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.
<u>Compensated Attenuator Cal Adjust</u>	Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.
<u>Get Misc Info</u>	Returns the text information set by the <u>Set Misc Info</u> VI of this library.
<u>Set Misc Info</u>	Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.
<u>Get Adjustment Date and Temperature</u>	Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.

Close Ext Cal Session	Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.
Subpalette	Description
Utilities	External calibration utilities for the PXIe-5113.

Open Ext Cal Session

Open Ext Cal Session

Opens an external calibration session for the specified device.


This VI returns an error if you specify a device that is not supported by this calibration library. This VI also downloads the calibration bitstream to the device FPGA if not already present.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name identifies the device to perform an external calibration on.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session returns a new external calibration session reference for the device specified by resource name. Use this reference as an input to other external calibration VIs in this library.

 error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Initialize

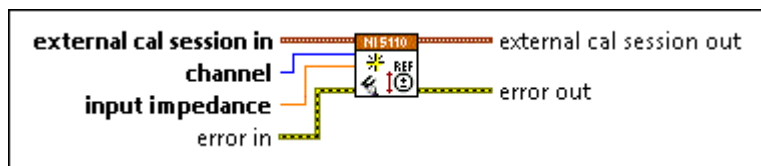
DC Reference Cal Initialize


Prepares the oscilloscope for adjusting the DC reference on the channel that you specify.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:


- PXle-5110
- PXle-5111
- PXle-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 channel specifies the input channel on which to perform the DC reference adjustment.

 input impedance specifies the input impedance to apply for the DC reference adjustment.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Configure

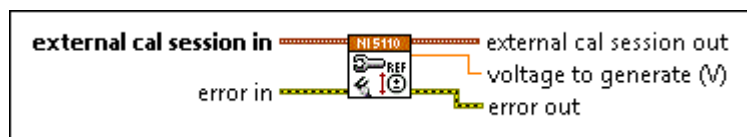
DC Reference Cal Configure


Configures the oscilloscope for the next DC reference calibration test point and waits for the oscilloscope to settle.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 voltage to generate (V) specifies the DC voltage to generate for the current test point.

 error out contains error information. This output provides [standard error out](#) functionality.

DC Reference Cal Adjust

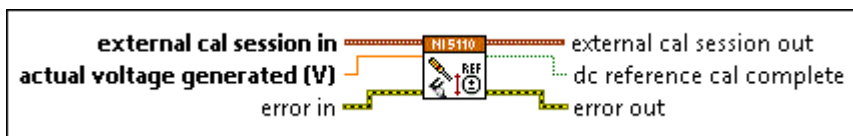
DC Reference Cal Adjust

Measures the input voltage on the specified channel, compares it with the actual voltage generated, performs an adjustment, and computes the calibration coefficients.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



EOG external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

DBL actual voltage generated (V) specifies the voltage applied to the channel under adjustment.

Note This voltage can vary by as much as $\pm 5\%$ from the requested voltage as defined by the voltage to generate parameter of [DC Reference Cal Configure](#). If the variation exceeds $\pm 5\%$, NI-SCOPE generates an error.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

EOG external cal session out passes a reference to your session to the next VI.

TF dc reference cal complete indicates whether the DC reference calibration is complete for the specified channel and the specified impedance.

ERR error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Initialize

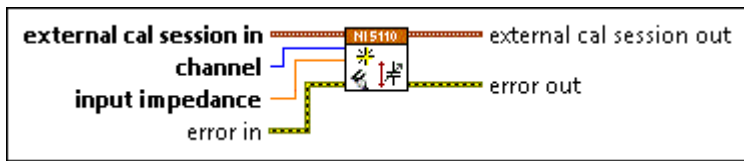
Compensated Attenuator Cal Initialize


Prepares the oscilloscope for adjusting the compensated attenuator on the channel you specify.


Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 channel specifies the input channel on which to perform the compensated attenuator adjustment.

 input impedance specifies the input impedance to apply for the compensated attenuator adjustment.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Configure

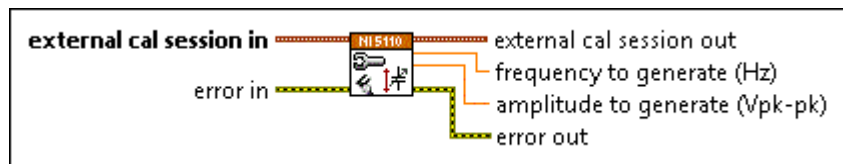
Compensated Attenuator Cal Configure

Configures the oscilloscope for the next compensated attenuator calibration test point and waits for the oscilloscope to settle.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

external cal session out passes a reference to your session to the next VI.

frequency to generate (Hz) specifies the frequency of the square waveform to generate for the current test point.

amplitude to generate (Vpk-pk) specifies the amplitude of the square waveform to generate for the current test point.

error out contains error information. This output provides [standard error out](#) functionality.

Compensated Attenuator Cal Adjust

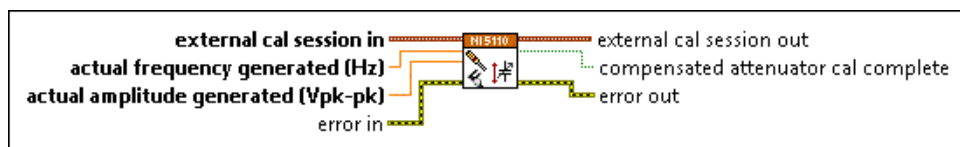
Compensated Attenuator Cal Adjust

Acquires the square waveform on the specified channel, calculates flatness error, performs an adjustment, and computes the calibration coefficients.

Refer to the calibration procedure for the oscilloscope you are calibrating for detailed instructions on the appropriate use of this VI.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



EOB external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session VI](#) of this library.




DBL actual frequency generated (Hz) specifies the frequency of the square waveform applied to the channel under adjustment.

Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the frequency to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.

DBL actual amplitude generated (Vpk-pk) specifies the amplitude of the square waveform applied to the channel under adjustment.

Note This frequency can vary by as much as $\pm 1\%$ from the requested frequency as defined by the amplitude to generate parameter of [Compensated Attenuator Cal Configure](#). If the variation exceeds $\pm 1\%$, NI-SCOPE generates as error.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

-  external cal session out passes a reference to your session to the next VI.
-  compensated attenuator cal complete indicates whether the compensated attenuator calibration is complete for the specified channel and the specified impedance.
-  error out contains error information. This output provides [standard error out](#) functionality.

Get Misc Info

Get Misc Info






Returns the text information set by the [Set Misc Info](#) VI of this library.

If you have not yet run the [Set Misc Info](#) VI during the calibration session, this VI will return the miscellaneous information that was last committed to the device.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



-  external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.
-  error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
-  external cal session out passes a reference to your session to the next VI.
-  miscellaneous information returns the text information.
-  error out contains error information. This output provides [standard error out](#) functionality.

Set Misc Info

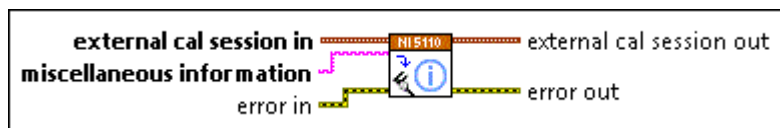
Set Misc Info


Stores custom text information that is stored until you close the current calibration session. You can choose to commit this information to the device when you close the calibration session.


Committing this information to the device allows you to access it in subsequent calibration sessions with the [Get Misc Info](#) VI of this library.


Supported Devices:


- PXle-5110
- PXle-5111
- PXle-5113




 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 miscellaneous information is the text information to store.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

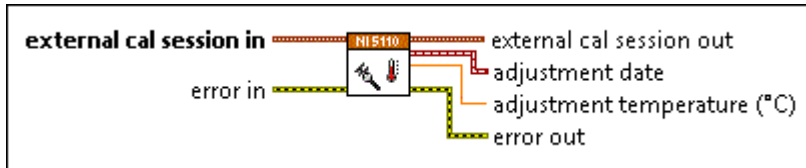
Get Adjustment Date And Temperature


Get Adjustment Date and Temperature


Returns the date of the last successful external calibration and the onboard temperature, in degrees Celsius, of the device at the time of that calibration.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 external cal session in identifies your session. This parameter is obtained from the [Open Ext Cal Session](#) VI of this library.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 external cal session out passes a reference to your session to the next VI.

 adjustment date returns the date and time of the last successful external calibration.

 adjustment temperature (° C) returns the temperature, in degrees Celsius, of the last successful external calibration.

 error out contains error information. This output provides [standard error out](#) functionality.

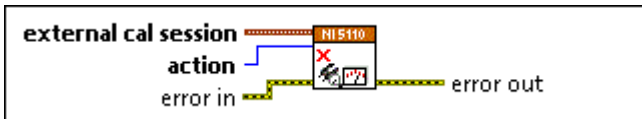
Close Ext Cal Session

Close Ext Cal Session

Closes the external calibration session. You can choose to either commit or disregard the results of the calibration session.

Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113



- external cal session is the external calibration session to close.
- action specifies the action to take if the external adjustment procedure completed without any errors or if the errors were corrected and the adjustment step was rerun using the same external calibration session.

Cancel (0)	Disregard the new calibration constants without changing any of the calibration data stored in the nonvolatile memory of the oscilloscope.
Commit (1)	Store the new calibration constants, adjustment time, adjustment date, and adjustment temperature in the nonvolatile memory of the oscilloscope.

Default Value: Cancel

- error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.
- error out contains error information. This output provides [standard error out](#) functionality.

Utilities

Utilities

Owning Palette: [5113 Ext Cal](#)
 External calibration utilities for the PXIe-5113.

PaletteObject	Description
Change External Cal Password	Changes the password used to begin an external calibration session.
Get External Cal Due Date	Returns the current external calibration due date of the device.

Set External Cal Due Date	Sets the external calibration due date of the device.
Get External Cal Verification Date	Returns the most recent external calibration verification date of the device.
Set External Cal Verification Date	Sets the most recent external calibration verification date of the device.

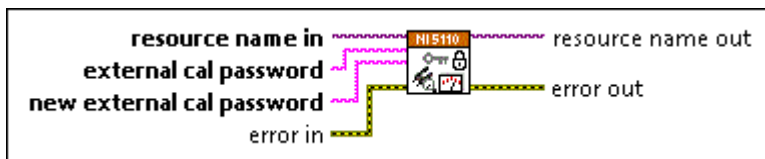
Change External Cal Password

Change External Cal Password


Changes the password used to begin an external calibration session.

Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 new external cal password is the new password to store in the EEPROM of the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Get External Cal Due Date

Get External Cal Due Date


Returns the current external calibration due date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113



 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal due date returns the date and time of the external calibration due date.

 error out contains error information. This output provides [standard error out](#) functionality.

Set External Cal Due Date

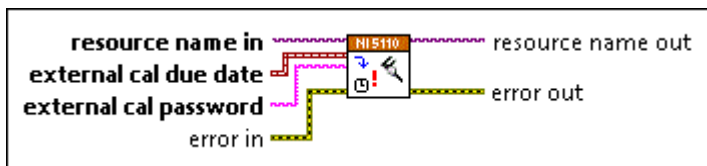
Set External Cal Due Date

Sets the external calibration due date of the device.

Supported Devices:

- PXIe-5110
- PXIe-5111

- PXIe-5113



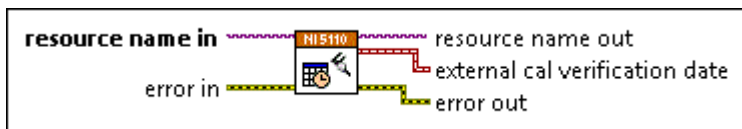
- resource name in identifies the device.
- external cal due date sets the new due date and time of the external calibration.
- external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.
The default password is NI.
- error in describes error conditions that occur before this node runs. This input provides standard error in functionality.
- resource name out passes the device name to the next VI.
- error out contains error information. This output provides standard error out functionality.


Get External Cal Verification Date


Returns the most recent external calibration verification date of the device.


Supported Devices:


- PXIe-5110
- PXIe-5111
- PXIe-5113




 resource name in identifies the device.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 external cal verification date returns the date and time of the most recent successful external verification of the device.

 error out contains error information. This output provides [standard error out](#) functionality.

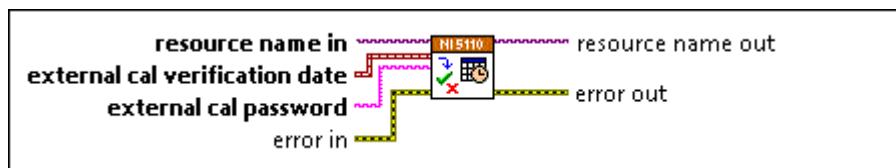
Set External Cal Verification Date

Set External Cal Verification Date


Sets the most recent external calibration verification date of the device.


Supported Devices:

- PXIe-5110
- PXIe-5111
- PXIe-5113





 resource name in identifies the device.


 external cal verification date sets the updated date and time of the most recent successful external verification of the device.

 external cal password is the calibration password for the device. This password is required to perform external calibration, set the calibration due date, and set the date and time of verification.

The default password is NI.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 resource name out passes the device name to the next VI.

 error out contains error information. This output provides [standard error out](#) functionality.

Measurements

Owning Palette: [NI-SCOPE VIs](#)

Use the NI-SCOPE Measurements VIs to perform waveform measurements, including scalar and array measurements.

PaletteObject	Description
niScope Clear Waveform Processing	Clears the list of processing steps assigned to the given channel.
niScope Add Waveform Processing	Adds one measurement to the list of processing steps that are completed before the measurement. The processing is added on a per channel basis, and the processing measurements are completed in the same order they are registered. All waveform measurements (for example, adding channels or applying a Bessel filter) are cached at the time of registering the processing, and this set of measurements is used during the processing step. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing happens before any other measurements.
niScope Fetch Measurement (poly)	Obtains a waveform from the digitizer and returns the specified measurement.
niScope Read Measurement	Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record.
niScope Clear Waveform Measurement Stats	Clears the waveform statistics on the channel and measurement you specify. This VI clears the s

	tatistical information and the multi-acquisition array measurements.
niScope Multi Read Measurement	Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for multiple channels and records.

niScope Add Waveform Processing

Adds one measurement to the list of processing steps that are completed before the measurement. The processing is added on a per channel basis, and the processing measurements are completed in the same order they are registered. All waveform measurements (for example, adding channels or applying a Bessel filter) are cached at the time of registering the processing, and this set of measurements is used during the processing step. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing happens before any other measurements.

For example, you can use a property node to set the NI-SCOPE property filter type to lowpass, and then register a Bessel Filter as a processing step using this VI. Then you can set the filter type to bandpass and register a Chebyshev filter. In a loop, call [niScope Read Measurement](#) with the scalar measurement function set to Voltage RMS. This process would repeatedly fetch a new waveform from the digitizer, perform the lowpass filter, perform the bandpass filter, and then compute the voltage RMS on the filtered waveform.


Refer to [Array Measurements](#) for more information about the available measurements.


Related topics:

- [Array Measurements in NI-SCOPE](#)
- [Scalar Measurements in NI-SCOPE](#)




 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 array measurement is the array measurement to add as a processing step.

Default Value: None

Refer to the list of NI-SCOPE [Array Measurements](#) for more information.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Fetch Measurement (poly)

Obtains a waveform from the digitizer and returns the specified measurement.

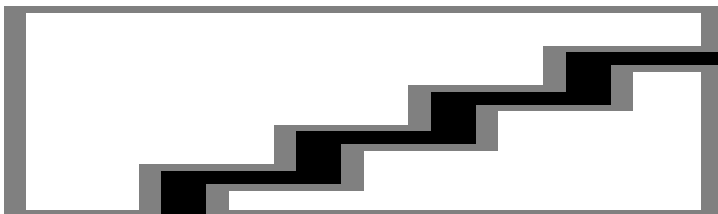
Refer to [Using Fetch VIs](#) for more information. Many of the measurements use the low, mid, and high reference levels. You configure the low, mid, and high references by using a property with [Chan Based Low Ref Level](#), [Chan Based Mid Ref Level](#), and [Chan Based High Ref Level](#) properties to set each channel differently.

Related topics


- [Making Waveform Measurements](#)


Measurement Cluster

Obtains a waveform from the digitizer and returns the specified measurement array for a single channel and record as a cluster along with timing data.




 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 array measurement is the array measurement to add as a processing step.

Default Value: None


Refer to the list of NI-SCOPE [Array Measurements](#) for more information.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.


Default Value: -1 (returns all available samples)


 **Note** Use the property [Fetch Meas Num Samples](#) to set the number of samples to fetch when performing a measurement. For more information about when to use this property with the niScope Fetch Measurement VI, refer to the NI KnowledgeBase.


 instrument handle out has the same value as the instrument handle.

 meas wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

FTB meas waveform returns a single record for single channel as a cluster for wiring to a graph, including the timing information.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

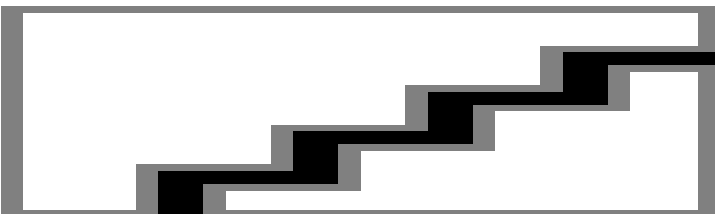
DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL measWfm returns a one-dimensional array of data for one waveform.


FTB error out contains error information. This output provides [standard error out](#) functionality.


Measurement 1D DBL

Obtains a waveform from the digitizer and returns the specified measurement voltage data for a single channel and record.




I/O instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 array measurement is the array measurement to add as a processing step.

Default Value: None


Refer to the list of NI-SCOPE [Array Measurements](#) for more information.


 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.


Default Value: -1 (returns all available samples)


 **Note** Use the property [Fetch Meas Num Samples](#) to set the number of samples to fetch when performing a measurement. For more information about when to use this property with the niScope Fetch Measurement VI, refer to the NI KnowledgeBase.


 instrument handle out has the same value as the instrument handle.

 meas wfm info contains all the timing and scaling information about the waveform.

 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

 absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

 relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

 xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

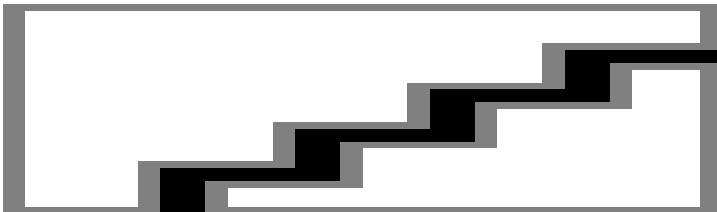
DBL reserved2 is reserved. Do not use.

DBL measWfm returns a one-dimensional array of data for one waveform.

Err error out contains error information. This output provides [standard error out](#) functionality.

Measurement 1D Cluster

Obtains a waveform from the digitizer and returns the specified array measurement for multiple channels and records as a cluster.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 array measurement is the array measurement to add as a processing step.

Default Value: None


Refer to the list of NI-SCOPE [Array Measurements](#) for more information.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

ERR error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

I32 meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.

Default Value: -1 (returns all available samples)

 Note Use the property Fetch Meas Num Samples to set the number of samples to fetch when performing a measurement. For more information about when to use this property with the niScope Fetch Measurement VI, refer to the NI KnowledgeBase.

I/O instrument handle out has the same value as the instrument handle.

ERR wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:


$$\mathbf{voltage} = \mathbf{binary\ data} \times \mathbf{gain\ factor} + \mathbf{offset}$$


DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

voltage = binary data × gain factor + offset

 reserved1 is reserved. Do not use.

 reserved2 is reserved. Do not use.

 meas waveform returns an array of clusters for wiring to a graph, including the timing information.


If you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first. For example, with a two-channel list, you would have the following index values:


index 0 = record 0, channel 0

index 1 = record 0, channel 1


index 2 = record 1, channel 0

index 3 = record 1, channel 1

 relativeInitialX is the initial x value relative to the trigger, such that the trigger always occurs at time equals zero.

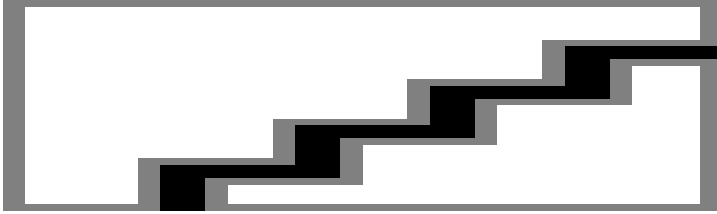
 xIncrement indicates the time in seconds between two samples in the acquired waveform.

 measWfm returns a one-dimensional array of data for one waveform.

 error out contains error information. This output provides [standard error out](#) functionality.

Measurement 2D DBL

Obtains a waveform from the digitizer and returns the specified measurement voltage data. Unlike the Measurement 1D DBL instance of the niScope Fetch Measurement VI, the Measurement 2D DBL instance can fetch data from multiple channels and records.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 array measurement is the array measurement to add as a processing step.

Default Value: None


Refer to the list of NI-SCOPE [Array Measurements](#) for more information.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I32 meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.

Default Value: -1 (returns all available samples)

 Note Use the property [Fetch Meas Num Samples](#) to set the number of samples to fetch when performing a measurement. For more information about when to use this property with the niScope Fetch Measurement VI, refer to the NI KnowledgeBase.

I/O instrument handle out has the same value as the instrument handle.

ERR wfm info contains all the timing and scaling information about the waveform.

I32 actualSamples is the actual number of samples in the acquired waveform; this number may be less than numSamples if the number you request is not available.

DBL absoluteInitialX is the timestamp in seconds of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

DBL relativeInitialX is the time in seconds from the trigger to the first sample in the acquired waveform.

DBL xIncrement indicates the time in seconds between two samples in the acquired waveform.

DBL offset is the offset factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL gain is the gain factor of the given channel.

Use for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

DBL reserved1 is reserved. Do not use.

DBL reserved2 is reserved. Do not use.

DBL wfm is an array of waveforms; that is, a two-dimensional array. This output can be wired directly to the LabVIEW waveform graph, but each waveform is plotted without timing information.

The 2D array includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal, there is one waveform per channel per record. If you call the fetch VI during a normal acquisition with the channel string "0,1" and the record number set to -1, the order of the output is:


record 0, channel 0

record 0, channel 1

record 1, channel 0

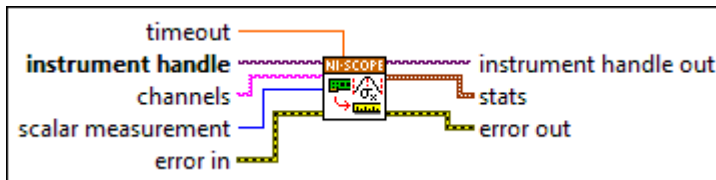
record 1, channel 1

The order of the channels is the order specified by the channels parameter.


 error out contains error information. This output provides [standard error out](#) functionality.


Statistics Cluster

Obtains a waveform measurement and returns the measurement value for a single channel and record. Specify a particular measurement type, such as rise time, frequency, or voltage peak-to-peak. The waveform on which the digitizer calculates the waveform measurement is from an acquisition that you previously initiated. The statistics for the specified measurement are also returned, where the statistics are updated once every acquisition when the specified measurement is fetched by any of the Fetch/Read measurement VIs.





 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 stats contains the resulting measurement data.

 result contains the measurement acquired.

DBL mean returns the mean scalar value, which is obtained by averaging each niScope Fetch Measurement Stats call.

DBL stdev returns the standard deviation.

DBL min returns the smallest scalar value acquired.

DBL max returns the largest scalar value acquired.

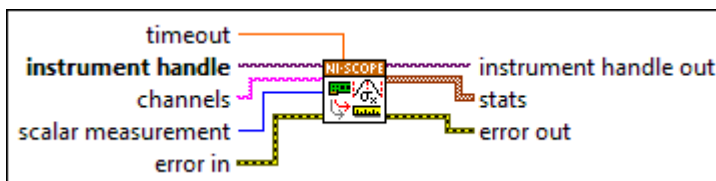
I32 num in stats returns the number of measurements used to calculate the statistics.

ERR error out contains error information. This output provides [standard error out](#) functionality.

Statistics 1D Cluster

Obtains a waveform measurement and returns the measurement value for multiple channel and multiple record acquisitions.

Specify a particular measurement type, such as rise time, frequency, or voltage peak-to-peak. The waveform on which the digitizer calculates the waveform measurement is from an acquisition that you previously initiated. The statistics for the specified measurement are also returned, where the statistics are updated once every acquisition when the specified measurement is fetched by any of the Fetch/Read waveform measurement VIs.





I/O instrument handle identifies a particular instrument session.


abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.


 stats is an array of clusters with scalar measurement results and statistics.

The array of clusters includes waveforms from multiple channels, records, and acquisition types. For example, if the acquisition type is normal there is one waveform per channel per record. If you call the multi fetch measurement VI during a normal acquisition with the channel list "0,1", the order of the results output is:


statistics for record 0, channel 0,
 statistics for record 0, channel 1,
 statistics for record 1, channel 0,
 statistics for record 1, channel 1

The order of the channels is the order specified by the channels parameter.

 scalar result contains the measurement.


 mean returns the mean scalar value, which is obtained by averaging each niScope Multi Fetch Measurement Stats call.

 stdev returns the standard deviation.

 min returns the smallest scalar value acquired.

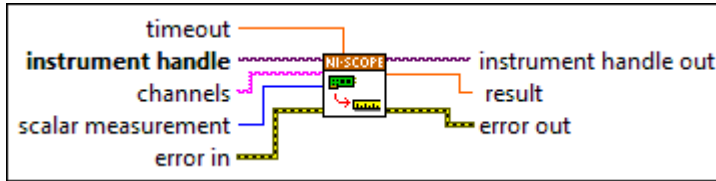
 max returns the largest scalar value acquired.

 num in stats returns the number of measurements used to calculate the statistics.

 error out contains error information. This output provides [standard error out](#) functionality.

Measurement Scalar DBL

Obtains a waveform from the digitizer and returns the specified measurement voltage data for a single channel and record.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

E= error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

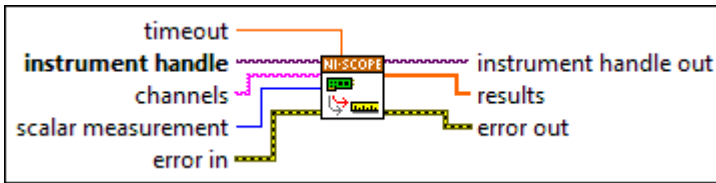
I/O instrument handle out has the same value as the instrument handle.

DBL result contains the measurement acquired.

E= error out contains error information. This output provides [standard error out](#) functionality.

Measurement Scalar 1D DBL

Fetches a waveform from the digitizer and performs the specified waveform measurement. Use this function for multiple channel and multiple record acquisitions.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

Err error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

DBL results contains the measurements acquired as a one-dimensional array; if you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.

For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

index 2 = record 1, channel 0

index 3 = record 1, channel 1


Err error out contains error information. This output provides [standard error out](#) functionality.


niScope Clear Waveform Processing


Clears the list of processing steps assigned to the given channel.


The processing steps are added with [niScope Add Waveform Processing](#), where the steps are completed in the same order in which they are registered. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing is also done before any other measurements.




 **instrument handle** identifies a particular instrument session.

 **channels** specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **instrument handle out** has the same value as the instrument handle.

 **error out** contains error information. This output provides [standard error out](#) functionality.

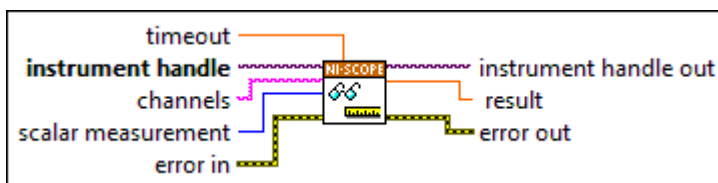
niScope Read Measurement

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record.

Use [niScope Multi Read Measurement](#) for multiple records and channels.

Related topics

- [Making Waveform Measurements](#)
- [Acquisition Functions](#)



 **instrument handle** identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

I32 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

DBL timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

ERR error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

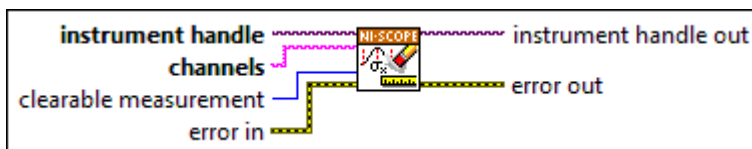
DBL result contains the measurement acquired.

ERR error out contains error information. This output provides [standard error out](#) functionality.

niScope Clear Waveform Measurement Stats

Clears the waveform statistics on the channel and measurement you specify. This VI clears the statistical information and the multi-acquisition array measurements.

Every time a measurement is called, the statistics information is updated, including the min, max, mean, standard deviation, and number of updates.




I/O instrument handle identifies a particular instrument session.


abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


I32 clearable measurement is the measurement for which to clear the statistics.

Refer to the list of NI-SCOPE [Scalar Measurements](#) or [Array Measurements](#) for constants. To clear all of the measurements, specify All Measurements.

Default Value: All Measurements

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

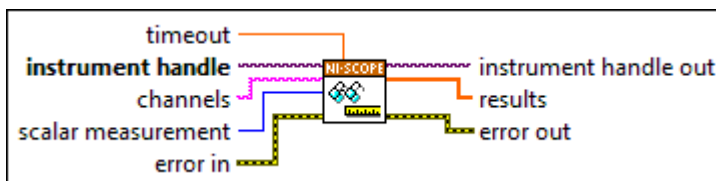
niScope Multi Read Measurement

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for multiple channels and records.


Use [niScope Read Measurement](#) for a single channel and record.


Related topics:

- [Making Waveform Measurements](#)
- [Acquisition Functions](#)





 instrument handle identifies a particular instrument session.


 channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).


 scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of NI-SCOPE [Scalar Measurements](#) for more information.

 timeout is the time (in seconds) to wait for the data to be acquired. Use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 results contains the measurements acquired as a one-dimensional array; if you specify a channel list, NI-SCOPE returns the waveforms in the list order. NI-SCOPE returns these records sequentially, so all record 0 waveforms are first.


For example, with a two-channel list, you would have the following index values:

index 0 = record 0, channel 0

index 1 = record 0, channel 1

index 2 = record 1, channel 0

index 3 = record 1, channel 1

 error out contains error information. This output provides [standard error out](#) functionality.

Utility Functions

Owning Palette: [NI-SCOPE VIs](#)

Use the NI-SCOPE Utility VIs to perform various tasks such as resetting the digitizer and returning the revision numbers of the driver and the instrument firmware.


PaletteObject	Description
niScope Reset	Resets the digitizer to its default state.
niScope Reset Device	Performs a hard reset of the device. Acquisition stops, all routes are released, RTSI and PFI lines are tristated, FPGAs are reset, hardware is configured to its default state, and all session attributes are reset to their default states.
niScope Disable	Aborts the current task, opens the data channel relays, and releases all RTSI and PFI lines. Use th

	is VI for a condition where you want to stop the current acquisition and disable the channels.
<u>niScope Probe Compensation Signal Start</u>	Generates a 1 kHz square wave signal for probe compensation.
<u>niScope Get Session Reference</u>	Extracts a session that can be passed to NI-TCI _k VIs. Session References are of generic type, which means that the corresponding wires are blue-green, unlike the wires for regular instrument driver sessions.
<u>niScope Error Message</u>	Takes the error code returned by NI-SCOPE VIs and returns the interpretation as a user-readable string.
<u>niScope Self Test</u>	Runs the device self-test routine and returns the test result(s).
<u>niScope Revision Query</u>	Returns the revision numbers of the instrument driver and instrument firmware.
<u>niScope Probe Compensation Signal Stop</u>	Disables the 1 kHz square wave signal for probe compensation.
<u>niScope Get Stream Endpoint Handle</u>	Returns a writer endpoint that can be used with NI-P2P to configure a peer-to-peer stream with a digitizer endpoint.
<u>niScope Is Device Ready</u>	Returns whether the device is ready to be used to the Device Ready? parameter.
<u>niScope CableSense Signal Start</u>	Generates the CableSense signal on all channels of an oscilloscope for which the signal is enabled, as configured by the <u>CableSense Mode</u> property.
<u>niScope CableSense Signal Stop</u>	Disables the CableSense signal on all channels of an oscilloscope for which the signal is enabled.
<u>niScope Get Channel Name (poly)</u>	Returns a comma-delimited list of channel names.

niScope Reset

Resets the digitizer to its default state.





 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Reset Device


Performs a hard reset of the device. Acquisition stops, all routes are released, RTSI and PFI lines are tristated, FPGAs are reset, hardware is configured to its default state, and all session attributes are reset to their default states.

Related Topics

- [Thermal Shutdown](#)





 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.


 instrument handle out has the same value as the instrument handle.


 error out contains error information. This output provides [standard error out](#) functionality.

niScope Disable

Aborts the current task, opens the data channel relays, and releases all RTSI and PFI lines. Use this VI for a condition where you want to stop the current acquisition and disable the channels.





 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

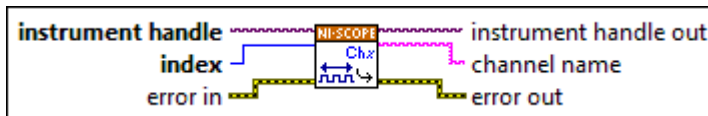
 error out contains error information. This output provides [standard error out](#) functionality.

niScope Get Channel Name (poly)


Returns a comma-delimited list of channel names.

niScope Get Channel Name


Returns the channel name(s) from a one-based index.




 instrument handle identifies a particular instrument session.

 index specifies a one-based index for the desired channel in the session.

Valid values are from one to the total number of channels in the session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 channel name returns a string of the channel name(s).

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Get Channel Name From String

Returns a comma-separated list of channel names from a string index list.




 instrument handle identifies a particular instrument session.


 index specifies an index list for the channels in the session.


Valid values are from zero to the total number of channels in the session minus one. The index string can be one of the following formats:


- A comma-separated list - for example, "0,2,3,1"
- A range using a hyphen - for example, "0-3"
- A range using a colon - for example, "0:3"

You can combine comma-separated lists and ranges that use a hyphen or colon. Both out-of-order and repeated indicies are supported ("2,3,0", "1,2,2,3"). White space characters, including spaces, tabs, feeds, and carriage returns, are allowed between characters. Ranges can be incrementing or decrementing.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 channel name returns a string of the channel name(s).

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Get Session Reference


Extracts a session that can be passed to NI-TClk VIs. Session References are of generic type, which means that the corresponding wires are blue-green, unlike the wires for regular instrument driver sessions.


Related topics:

- [NI-TClk Overview](#)




 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 session reference references the device session that can be passed to NI-TClk VIs.

 error out contains error information. This output provides [standard error out](#) functionality.


niScope Get Stream Endpoint Handle


Returns a writer endpoint that can be used with NI-P2P to configure a peer-to-peer stream with a digitizer endpoint.


Related topics:


- [Peer-to-Peer Streaming](#)





 instrument handle identifies a particular instrument session.

 stream endpoint is the stream endpoint FIFO to configure. Refer to the device-specific documentation for peer-to-peer streaming in the **High-Speed Digitizers Help** for more information.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 writer handle is a reference to a peer-to-peer writer FIFO that can be used to create a peer-to-peer streaming session.


 error out contains error information. This output provides [standard error out](#) functionality.

niScope Error Message

Takes the error code returned by NI-SCOPE VIs and returns the interpretation as a user-readable string.

VI_NULL can be passed as the instrument handle, which is useful to interpret errors if [niScope Initialize](#) has failed.



 instrument handle identifies a particular instrument session.

 error code (0) is the status code returned by any NI-SCOPE VI.

 error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error message returns the interpreted error code as a user-readable string.

 error out contains error information. This output provides [standard error out](#) functionality.


niScope Is Device Ready

Returns whether the device is ready to be used to the Device Ready? parameter.

Related topics

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)



 resource name specifies the device name assigned by Measurement & Automation Explorer (MAX).

Examples


Example	Device Type	Syntax
1	NI-DAQmx device	myDAQmxDevice (myDAQmx Device = device name)
2	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::2 (2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)


For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternative syntax for NI-DAQmx devices consists of DAQ::NI-DAQmx device name, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if


the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.


You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

 Note NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

 **Device Ready?** Returns whether the device is ready to use.

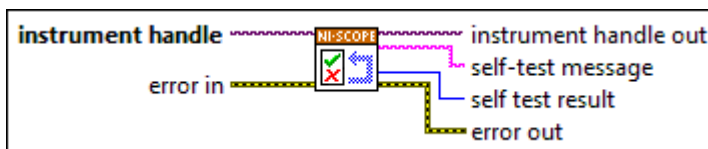
Default Value: None

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.


 **error out** contains error information. This output provides [standard error out](#) functionality.


niScope Self Test

Runs the device self-test routine and returns the test result(s).




 **instrument handle** identifies a particular instrument session.

 **error in** describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 **instrument handle out** has the same value as the instrument handle.

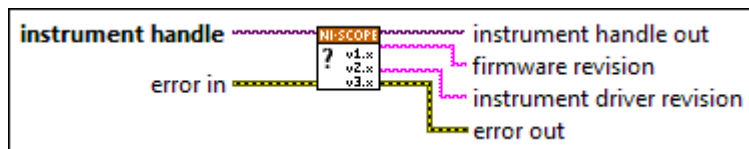
 **self-test message** returns the self-test response string from the device.

 **self test result** contains the value returned from the device self-test.


 error out contains error information. This output provides [standard error out](#) functionality.

niScope Revision Query

Returns the revision numbers of the instrument driver and instrument firmware.





 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 firmware revision returns the instrument firmware revision numbers.

 instrument driver revision returns the instrument driver software revision numbers.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Probe Compensation Signal Start


Generates a 1 kHz square wave signal for probe compensation.

Most oscilloscopes output the probe compensation signal on PFI 1.

The following oscilloscopes output the probe compensation signal in unique locations.


Device	Output Location	Notes
PXIe-5110 PXIe-5111 PXIe-5113	Probe compensation terminal	The signal at this terminal is enabled by default.
PXIe-5163 PXIe-5164	SMB PFI 0	Though the PFI 0 line is also available from the AUX 0 MHDMR connector of these oscilloscopes, the probe compensation signal is output from the probe compensation terminal.

l is available only from SMB PFI 0.

 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.



 instrument handle identifies a particular instrument session.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.


niScope Probe Compensation Signal Stop

Disables the 1 kHz square wave signal for probe compensation.


 Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.



 instrument handle identifies a particular instrument session.


 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope CableSense Signal Start

Generates the CableSense signal on all channels of an oscilloscope for which the signal is enabled, as configured by the [CableSense Mode](#) property.

 Note The input impedance of the channel(s) to convey the CableSense signal must be set to 50 Ω .


You can call this VI only during an acquisition. If you call this VI while your oscilloscope is not acquiring, NI-SCOPE generates an error.


Supported Devices


- [PXIe-5110](#)
- [PXIe-5111](#)
- [PXIe-5113](#)
- [PXIe-5160](#)
- [PXIe-5162](#)



 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope CableSense Signal Stop


Disables the CableSense signal on all channels of an oscilloscope for which the signal is enabled.


Supported Devices


- [PXIe-5110](#)
- [PXIe-5111](#)
- [PXIe-5113](#)
- [PXIe-5160](#)
- [PXIe-5162](#)



 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

NI-SCOPE Express (Deprecated)

This VI is deprecated. NI recommends that you use the niScope EX Configured Acquisition.vi example instead of this VI as a starting point for new applications you create. This VI will execute in code created with versions of NI-SCOPE earlier than 19.0, but it cannot be reconfigured from a dialog box with NI-SCOPE 19.0 or later. To reconfigure this VI, right-click the Express VI and select Open Front Panel»Convert. This will convert the Express VI to code that you can configure on the block diagram.

Acquires an analog voltage waveform from an NI high-speed digitizer.

Hardware Support

The NI-SCOPE Express VI does not support the following hardware:

- PXIe-5110/5111/5113
- PXIe-5163
- PXIe-5164
- PXIe-5170

- PXIe-5171
- PXIe-5172

[Dialog Box Options](#)

[Block Diagram Inputs](#)

[Block Diagram Outputs](#)

[Examples](#)

Dialog Box Options

Parameter	Description
Autoscale graph	Enables and disables autoscaling on the Y-axis of the Acquired Signals graph.
Configuration	<p>Contains the following options:</p> <ul style="list-style-type: none"> ▪ Device—Specifies the device used. Unavailable devices are disabled. ▪ Auto Setup—Automatically configures the device based on the current inputs. ▪ Per Div Mode—A Boolean that enables or disables per division views. When enabled, Range (V/Div) and Time/Division are visible. ▪ Vertical—Configures the channel-based parameters for the selected (highlighted) channel. Displays the following parameters: <ul style="list-style-type: none"> ▪ Channels—Select or multi-select channel(s) by highlighting. The channel configurations for selected channels are displayed. Place a checkmark next to the channel(s) you want to enable. Only enabled channels will acquire data. ▪ Range (V)—Sets the full vertical range in volts. ▪ Range (V/Div)—When Per Div Mode is enabled, sets the vertical range in volts per division.

- Offset (V)—Sets the voltage offset.
- Probe attenuation—Sets the probe atenuation.
- Coupling—Sets the coupling on the analog signal.
- Input impedance—Sets the input impedance.
- Bandwidth (Hz)—Sets the frequency of the input lowpass filter.
- Horizontal—Configures the horizontal options.
 - Enable TIS—Enables time interleaved sampling (TIS), which extends the maximum sample rate on the specified channel for devices that support TIS.
 - Sample rate (S/s)—Specifies the digitizing rate in samples/sec.
 - Time/Division—When Per Div Mode is enabled, sets the digitizing rate in time per division.
 - RIS Enabled—If lit, indicates that Random Interleaved Sampling is enabled. This occurs when the Sample rate (S/s) is higher than the maximum real-time sample rate of the device.
 - Record/Read length—Number of data points acquired from the digitizer at each run of the VI.
 - Acquire—Sets the acquisition mode of the block. Options are:
 - N Samples—Acquire a finite number of data points.
 - Continuously—Acquire phase-continuous data.

Trigger

Configures the trigger options. The visible options change depending upon the value of Type. Contains the following:

- **Type**—Sets the trigger type. All listed types may not be supported by all devices.
- **Ref position (%)**—Sets the trigger position in the acquisition as a percent of the full length of the acquisition.
- **Max time (s)**—Sets the maximum time each call of this VI will take. This is a timeout.
- **Source**—Sets the trigger source. The options change depending upon Type.
- **Level (V)**—Sets the trigger level.
- **Hysteresis (V)**—Sets the trigger hysteresis.
- **Low level (V)**—Sets the window-mode trigger low level.
- **High level (V)**—Sets the window-mode trigger high level.
- **Slope**—Sets the trigger slope.
- **Window mode**—Sets the window triggering mode.
- **Holdoff (s)**—Sets the minimum time between trigger events.
- **Delay (s)**—Sets the time between the actual trigger event and the Ref position (%) in the data.
- **Coupling**—Sets the trigger path coupling. If the trigger channel is also a data channel, this option may be modified by the channel coupling.
- **Signal format**—Sets the video format for video triggering.
- **DC restore**—Enables and disables video triggering DC restore.

	<ul style="list-style-type: none"> ■ Polarity—Sets the video triggering polarity. ■ Event—Sets video trigger event. ■ Line number—Sets the video trigger line number used when Event is set to Line Number.
Advanced	<p>Configures options used less frequently. Contains the following:</p> <ul style="list-style-type: none"> ■ Reference clock source—Sets the location for locking the reference clock of the device to an external source. ■ Use relative initial time—If checked, the initial time in the output waveform is set so that the trigger position (reference position) is at zero time. If unchecked, the initial time is the absolute time the waveform was acquired, as accurately as the hardware can determine it.
Execution Control	<p>Contains the following options:</p> <ul style="list-style-type: none"> ■ Pre-execution delay (ms)—Specifies the amount of time to wait before the step executes. If you configure the step to start after another step, the delay represents the amount of time to wait after the Step to wait for has started. ■ Post-execution delay (ms)—Specifies the amount of time to wait after the step executes. ■ Start this step after—Use this control to make the step wait until another step has started before executing. You can make this step wait on any other hardware step in the project by using the Step to wait for ring control. You can use this control to force an acquisition device to start after a generation device has started. You can also use this control to ensure that a device ge

nerating a trigger signal starts after the device receiving the signal, to avoid sending the signal before the receiver is ready.

- Step to wait for—Lists the possible hardware steps for which this step can wait.

Block Diagram Inputs

Parameter	Description
Close	Used for optimization in a loop. If TRUE, close all references and restart the device the next time called. If FALSE, use the current device handle on the next call. The default varies depending upon the value of Acquire. The default is appended to the control name.
Max Time	Specifies the timeout value for the acquisition.
error in	Describes error conditions that occur before this VI or function runs.

Block Diagram Outputs

Parameter	Description
Signal	Returns the data collected by the digitizer. The number of channels depends upon how many channels are enabled.
error out	Contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces.

Examples

Refer to the examples in the labview\examples\instr\niScope directory or use the NI Example Finder in LabVIEW (Help»Find Examples) for more examples of how to use NI-SCOPE.

niScope Actual Meas Wfm Size

Returns the total available size of an array measurement acquisition in samples.



I/O instrument handle identifies a particular instrument session.

I32 arrayMeasFunction is the measurement to perform.

Refer to the list of NI-SCOPE [Array Measurements](#) for more information.

E- error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

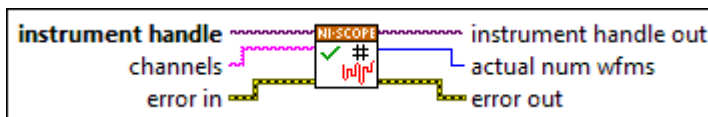
I/O instrument handle out has the same value as the instrument handle.

I32 meas wfm size returns the size of the resulting analysis waveform in samples.

E- error out contains error information. This output provides [standard error out](#) functionality.

niScope Actual Num Wfms

Allows you to declare appropriately sized waveforms. NI-SCOPE handles the channel list parsing for you.



I/O instrument handle identifies a particular instrument session.

abc channels specifies the channel(s) from which to acquire data. For more information, refer to [Channel String Syntax](#).

E- error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

I/O instrument handle out has the same value as the instrument handle.

U32 actual num wfms returns the number of records times the number of channels; if the acquisition Type parameter of the [niScope Configure Acquisition](#) VI is set to DDC, this value is multiplied by two.

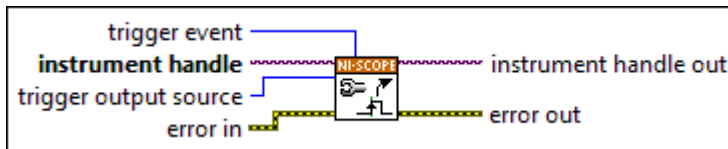
Err error out contains error information. This output provides [standard error out](#) functionality.

niScope Configure Trigger Output

Configures the digitizer to generate a signal pulse that other digitizers can detect when configured for digital triggering.

For NI-DAQmx devices, closing the session clears the routes. However, if you want to clear the routes before closing the session, call this VI again, routing the Stop Trigger to None.

This VI is obsolete. Consider using [niScope Export Signal](#) instead.



I/O instrument handle identifies a particular instrument session.

U16 trigger output source specifies the hardware signal line on which the digital pulse is generated.

Default Value: No event (none)

Defined Values

None

RTSI 0


RTSI 1


RTSI 2


RTSI 3


RTSI 4

RTSI 5
 RTSI 6
 RTSI 7
 PFI 0
 PFI 1
 PFI 2
 PXI Star

 trigger event specifies the condition in which this device will generate a digital pulse.

 error in describes error conditions that occur before this node runs. This input provides [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Reset with Defaults

Resets the device to the default state and applies any initial default settings from the IVI Configuration Store. This VI uses default parameters to do a software reset on the device. The changes are not immediately committed to hardware.




 instrument handle identifies a particular instrument session.

 error in describes error conditions that occur before this node runs.

The default is `no error`. This node executes regardless of an incoming error.

This input contains status, code, and source, which provide [standard error in](#) functionality.

 instrument handle out has the same value as the instrument handle.

 error out contains error information. This output provides [standard error out](#) functionality.

niScope Properties

May 2020

Use the NI-SCOPE properties to access advanced configuration options for high-speed digitizer applications.

Property	Description
Active Channel	Specifies the channel name used to access all subsequent properties in this instance of the property node. If the property you want to use is Channel Based, you need to first select this property and then pass the name of the specific channel. If the property you specify is not channel based, pass an empty string, or omit setting this property. The default value is "". Details
Vertical:Channel Enabled	Specifies whether the digitizer acquires a waveform for the channel. Details
Vertical:Vertical Range	Specifies the absolute value of the input range for a channel. The units are volts. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V. Details
Vertical:Vertical Offset	Specifies the location of the center of the range. The value is with respect to ground and is in volts. For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this property to 5.0 V. This property is not supported by all digitizers. Details
Vertical:Maximum Input Frequency	Specifies the bandwidth of the channel in hertz. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB. Details
Vertical:Vertical Coupling	Specifies how the digitizer couples the input signal for the channel. When you change input coupling, the input stage takes a finite amount of time to settle. Details

Vertical:Input Impedance	Specifies the input impedance for the channel in ohms. Details
Vertical:Probe Attenuation	Specifies the probe attenuation for the input channel. For example, for a 10:1 probe, you would set this property to 10.0. Details
Vertical:Channel Terminal Configuration	Specifies how the digitizer configures the channel terminal. Details
Vertical:Advanced:Digital Gain	<p>Applies gain to the specified channel in hardware before any onboard signal processing occurs. The default value is 1.</p> <p>The output of the digital gain/offset block is as follows:</p> <p>(ADC value × digital gain) + digital offset</p> <p>Units: Unitless</p> <p>Valid Values: -1.5 to 1.5 Details</p>
Vertical:Advanced:Digital Offset	<p>Applies offset to the specified channel in hardware before any onboard signal processing occurs. The default value is 0.</p> <p>Units: Volts</p> <p>Valid Values</p> <p>$\pm(\text{Vertical Range} \times 0.4)$ Details</p>
Vertical:Advanced:Bandpass Filter Enabled	Enables the bandpass filter on the specified channel. For the NI PXIe-5622, set the value to TRUE to enable the IF filtered path 50MHz bandpass filter centered at 187MHz. The default value is FALSE. Details
Vertical:Advanced:Dither Enabled	Enables or disables the analog dither on the device. Using dither can improve the spectral performance of the device by reducing the effects of quantization. However, adding dither increases the power level to the ADC, so you may need to either decrease the signal level or increase the vertical range. The default value is FALSE. Details

Vertical:Advanced:Flex FIR Antialias Filter Type	The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter. Use this property to select from several types of filters to achieve desired filtering characteristics. For most applications, the default value of this property is recommended. The other available filters are useful for optimizing settling time measurements of step responses. The default value is 48 Tap Standard. Details
Vertical:Advanced:High Pass Filter Frequency	Specifies the frequency for the highpass filter in Hz. The device uses one of the valid values listed below. If an invalid value is specified, no coercion occurs. The default value is 0. (PXIe-5164) Valid Values: <ul style="list-style-type: none"> ▪ 0 ▪ 90 ▪ 450
Vertical:Advanced:Interleaving Offset Correction Enabled	Enables the interleaving offset correction on the specified channel. The default value is TRUE. Details
Horizontal:Min Sample Rate	Specifies the sampling rate (in Samples/second) for the acquisition. This attribute is invalid when the device is configured to use an external sample clock timebase. When a DDC is enabled, this attribute specifies the IQ rate. When both the Time Per Record Property and the Min Sample Rate Property are set, the attribute that was set first is ignored. Details
Horizontal:Min Number of Points	Specifies the minimum number of points you require in the waveform record for each channel. Details
Horizontal:Number of Records	Specify the number of records to acquire. Details

Horizontal:Reference Position	Specifies the position of the Reference Event in the waveform record as a percentage of the record. Details
Horizontal:Actual Sample Rate	Returns the actual sample rate used for the acquisition. Units: hertz (Samples / Second) Details
Horizontal:Actual Record Length	Returns the actual number of points the digitizer acquires for each channel. The value is equal to or greater than the value you specify in the niScope Configure Horizontal Timing VI . Valid Values: 1 to the maximum memory size Details
Horizontal:Enable Time Interleaved Sampling	Extends the maximum sample rate on the specified Active Channel for some devices that support Time Interleaved Sampling (TIS). TIS enables the device to use multiple ADCs to sample the same waveform at a higher effective real-time rate. NI 5152/5153/5154 devices fully support Read/Write ability for this property. For other devices that use TIS mode, such as the NI 5185/5186, this property is Read Only. Details
Horizontal:Enforce Realtime	Indicates whether the digitizer enforces real-time measurements or allows equivalent-time measurements. Details
Horizontal:Enable Records > Memory	Allows you to acquire more records than fit in onboard memory. TRUE—Enables NI-SCOPE to fetch more records than fit in memory FALSE—Disables NI-SCOPE from fetching more records than fit in memory Details
Horizontal:RIS Num Avg	Specifies the number of averages in each RIS bin. Details

Horizontal:RIS Method	Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the Max Realtime Sample Rate . Details
Horizontal:Maximum Real Time Sample Rate	Returns the maximum real-time sample rate in hertz. Details
Horizontal:Maximum RIS Rate	Returns the maximum RIS sampling rate in hertz. Details
Horizontal:Memory Size	Returns the total combined amount of onboard memory for all channels in bytes. Details
Horizontal:Advanced:Enable TDC	Specifies that the digitizer should record the trigger position precisely using time-digital conversion (TDC). Details
Horizontal:Advanced:Time Per Record	Specifies the length of time (in seconds) that corresponds to the record length. This attribute is invalid when the device is configured to use an external sample clock timebase. This attribute is also invalid when a DDC is enabled. When both the Time Per Record Property and the Min Sample Rate Property are set, the attribute that was set first is ignored. Details
Horizontal:Advanced:Acquisition Start Time	Specifies the length of time (in seconds) from the trigger event to the first point in the waveform record. Details
Triggering:Trigger Type	Specifies the type of trigger to use. Details
Triggering:Trigger Source	Specifies the source the digitizer monitors for the trigger event. The value must be selected from one of the following valid values. Details
Triggering:Trigger Output Terminal	Specifies the destination to export the Reference (Stop) Trigger Refer to the device specifications document for a list of valid destinations. Details
Triggering:Terminal Name	Returns the fully qualified name for the Reference Trigger terminal. Details

Triggering:Trigger Level	Specifies the voltage threshold for the trigger. The units are volts. Details
Triggering:Trigger Modifier	Configures the device to automatically complete an acquisition if a trigger has not been received. Details
Triggering:Auto Triggered	Specifies whether the acquisition was triggered automatically. Auto triggering occurs if the Trigger Modifier property is set to Auto Trigger and no trigger has been received for a certain amount of time. Details
Triggering:Trigger Hysteresis	Specifies the size of the hysteresis window, in volts, on either side of the trigger level. Details
Triggering:Trigger Delay	Specifies the trigger delay time in seconds. Details
Triggering:Trigger Holdoff	Specifies the length of time the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger. The units are seconds. Details
Triggering:Trigger Slope	Specifies whether a rising or a falling edge triggers the digitizer. Details
Triggering:Trigger Coupling	Specifies how the digitizer couples the trigger source. Details
Triggering:Trigger Impedance	Sets the impedance for the external trigger input. Details
Triggering:Start To Ref Trigger Holdoff	Pass the length of time (in seconds) you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a reference (stop) trigger. Details
Triggering:End of Record to Advance Trigger Holdoff	End of Record to Advance Trigger Holdoff is the length of time (in seconds) that a device waits between the completion of one record and the acquisition of pre-trigger samples for the next record. During this time, the acquisition engine state delays the transition to the Wait for Advance Tri

	<p>gger state, and will not store samples in onboard memory, accept an Advance Trigger, or trigger on the input signal..</p> <p>Supported Devices: NI 5185/5186 Details</p>
Triggering:Trigger Window:Window Mode	Specifies whether to trigger when the signal enters or leaves the window specified by the Trigger Window Low Level property or the Trigger Window High Level property. Details
Triggering:Trigger Window:Low Level	Pass the lower voltage threshold you want the digitizer to use for window triggering. Details
Triggering:Trigger Window:High Level	Pass the upper voltage threshold you want the digitizer to use for window triggering. Details
Triggering:Trigger Video:Signal Format	Specifies the video signal format to use. Details
Triggering:Trigger Video:Line Number	Specifies the line number to trigger on. Details
Triggering:Trigger Video:Polarity	Specifies whether the video signal is positive or negative. Details
Triggering:Trigger Video:Event	Specifies the event to trigger on. Details
Triggering:Trigger Video:Enable DC Restore	Restores the video-triggered data retrieved by the digitizer to the video signal's zero reference point. The default value is FALSE. Details
Triggering:Trigger Glitch:Glitch Condition	Specifies whether the oscilloscope triggers on pulses of duration less than or greater than the specified Glitch Width . Details
Triggering:Trigger Glitch:Glitch Width	Specifies the glitch duration, in seconds. Details
Triggering:Trigger Glitch:Glitch Polarity	Specifies the polarity of pulses that trigger the oscilloscope for glitch triggering. Details
Triggering:Trigger Width:Width Condition	Specifies whether the oscilloscope triggers on pulses within or outside the duration range bounded by the Width Low Threshold and Width High Threshold properties. Details

Triggering:Trigger Width:Width Low Threshold	Specifies the low width threshold, in seconds. Details
Triggering:Trigger Width:Width High Threshold	Specifies the high width threshold, in seconds. Details
Triggering:Trigger Width:Width Polarity	Specifies the polarity of pulses that trigger the o scilloscope for width triggering. Details
Triggering:Trigger Runt:Runt Low Threshold	Specifies the lower of two thresholds, in volts, t hat bound the vertical range to examine for runt pulses. Details
Triggering:Trigger Runt:Runt High Threshold	Specifies the higher of two thresholds, in volts, t hat bound the vertical range to examine for runt pulses. Details
Triggering:Trigger Runt:Runt Polarity	Specifies the polarity of pulses that trigger the o scilloscope for runt triggering. Details
Triggering:Trigger Runt:Runt Time Condition	Specifies whether runt triggers are time qualifie d, and if so, how the oscilloscope triggers in rela tion to the duration range bounded by the Runt Time Low Limit and Runt Time High Limit prope rties. Details
Triggering:Trigger Runt:Runt Time Low Limit	Specifies, in seconds, the low runt threshold tim e. Details
Triggering:Trigger Runt:Runt Time High Limit	Specifies, in seconds, the high runt threshold ti me. Details
Device:CableSense:Voltage	Returns the voltage of the CableSense signal tha t is written to the EEPROM of the oscilloscope d uring factory calibration. Details
Device:CableSense:Mode	Specifies whether and how the oscilloscope is c onfigured to generate a CableSense signal on th e specified channels when the niScope CableSense Signal Start VI is called. Details

Acquisition:Enabled Channels	Returns a comma-separated list of the channels enabled for the session in ascending order. Details
Triggering:Onboard Signal Processing:Ref Trigger Detection Location	Specifies which reference trigger detection circuitry to use on the device. Details
Triggering:Onboard Signal Processing:Ref Trigger Min Quiet Time	Specifies the amount of time (in seconds) the trigger circuit must not detect a signal above the trigger level (or below the trigger level if the trigger slope is negative) before the trigger is armed. This property is useful for triggering at the beginning of signal bursts instead of in the middle of signal bursts. The default value is 0. Details
Fetch:Fetch Record Number	Sets the record to fetch. The record is from a channel you specify. The default value is 0. Valid Values: Values greater than or equal to 0 Details
Fetch:Fetch Number of Records	Fetches multiple records. If you want to fetch all records from the record you specify in the Fetch Record Number property to the last record configured, use -1. The default value is -1. Details
Fetch:Fetch Relative To	Specifies which point in the acquired waveform is the first to be fetched. This property specifies what the 'Fetch Offset' is relative to. Details
Fetch:Fetch Offset	Sets the offset in samples; the samples returned also depend on the Fetch Relative To property. The default value is 0. Valid Values: All integers Details
Fetch:Data Transfer Block Size	Specifies the maximum number of samples to transfer at one time from the device to host memory. Increasing this number should result in better fetching performance because the driver does not need to restart the transfers as often. However, increasing this number may also increase the amount of page-locked memory required from the system. Details

Fetch:Fetch Meas Num Samples	Determines the number of samples to fetch from a digitizer when performing a measurement. -1 means fetch all samples from the Fetch Offset property to the end of the current record. The default value is -1. Details
Fetch:Points Done	Actual number of samples acquired relative to the configured value for Fetch Relative To , including Fetch Offset , and for the current configured Fetch Record Number . Details
Fetch:Fetch Backlog	Specifies the number of points acquired that have not been fetched yet. Details
Fetch:Records Done	Returns the number of records your digitizer has acquired. Details
Fetch:Advanced:Maximum Bandwidth	Specifies the maximum bandwidth that the device is allowed to consume. The NI device limits itself to transfer fewer bytes per second on the PCIe bus than the value you specify for this property. Details
Fetch:Advanced:Preferred Packet Size	Specifies the preferred size of the data field in the PCI Express packet. In general, the larger the packet size, the more efficiently the device uses the bus. However, some systems, because of their implementation, perform better with smaller packet sizes. The value of this property must be a power of two (64, 128, ... , 512). Details
Clocking:Reference (Input) Clock Source	Specifies the input source for the PLL reference clock. Defined Values VAL_NO_SOURCE VAL_PFI_0 VAL_PFI_1 VAL_PFI_2 VAL_PXI_CLOCK

	<p>VAL_CLK_IN</p> <p>VAL_EXTERNAL</p> <p>VAL_RTSI_CLOCK Details</p>
Cloning:Reference Clock Rate	If Reference Clock Source is an external source, specifies the frequency (in hertz) of the input clock (reference clock) to which the internal sample clock timebase is synchronized. Details
Cloning:Output Clock Source	Specifies the output source for the 10 MHz clock to which the sample clock of another digitizer can be phase-locked. Details
Cloning:Sample Clock Timebase Source	Specifies the source of the sample clock timebase, which is the timebase used to control waveform sampling. Details
Cloning:Sample Clock Timebase Rate	Specifies, in hertz, the frequency of the external clock used as the timebase source if the Sample Clock Timebase Source is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock. Details
Cloning:Sample Clock Timebase Divisor	If Sample Clock Timebase Source is an external source, specifies the ratio between the sample clock timebase rate and the actual sample rate, which can be slower. Details
Cloning:Sample Clock Timebase Multiplier	If Sample Clock Timebase Source is an external source, this property specifies the ratio between the Sample Clock Timebase Rate and the actual sample rate, which can be higher. This property can be used in conjunction with the Sample Clock Timebase Divisor Property . Details
Cloning:Exported Sample Clock Output Terminal	Exports the sample clock to a specified terminal. This property is not supported by all digitizers. Details

Clocking:Clock Sync Pulse Source	Specifies the line on which the sample clock or the one-time sync pulse is sent or received. Details
Clocking:PLL Lock Status	If TRUE, the PLL has remained locked to the external reference clock since it was last checked. If FALSE, the PLL has become unlocked from the external reference clock since it was last checked. Details
Clocking:Advanced:Oscillator Phase DAC Value	Gets or sets the binary phase DAC value that controls the delay added to the phase-locked loop (PLL) of the sample clock. Details
Clocking:Advanced:Absolute Sample Clock Offset	Gets or sets the absolute time offset, in seconds, of the sample clock relative to the reference clock. Details
Device:FPGA Bitfile Path	Gets the absolute file path to the bitfile loaded on the FPGA. Details
Synchronization:Start Trigger (Acq. Arm): Source	Specifies the source the digitizer monitors for an acquisition arm trigger. When an acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples. Details
Synchronization:Start Trigger (Acq. Arm): Output Terminal	Specifies the destination to export the Start trigger. When the start trigger is received, the digitizer begins acquiring data. Refer to the device specifications document for a list of valid destinations. Details
Synchronization:Start Trigger (Acq. Arm):Terminal Name	Returns the fully qualified name for the Start Trigger terminal. Details
Synchronization:Ready for Start:Output Terminal	Specifies the destination to export the Start trigger. When the start trigger is received, the digitizer begins acquiring data. Refer to the device specifications document for a list of valid destinations. Details
Synchronization:Ready for Start:Terminal Name	Returns the fully qualified name for the Ready for Start Event terminal. Details

Synchronization:Advance Trigger:Source	Specifies the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples for the next record. Details
Synchronization:Advance Trigger:Output Terminal	Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples. Details
Synchronization:Advance Trigger:Terminal Name	Returns the fully qualified name for the Advance Trigger terminal. Details
Synchronization:Ready for Advance:Output Terminal	Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples. Details
Synchronization:Ready for Advance:Terminal Name	Returns the fully qualified name for the Ready for Advance Event terminal. Details
Synchronization:Arm Reference Trigger:Source	Specifies the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins searching for the reference (stop) trigger from the user-configured trigger source. Details
Synchronization:Ready for Reference:Output Terminal	Specifies the destination for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger. Refer to the device-specific documentation in the NI High-Speed Digitizers Help for a list of valid destinations for your device. Details
Synchronization:Ready for Reference:Terminal Name	Returns the fully qualified name for the Ready for Reference Event terminal. Details
Synchronization:End of Record:Output Terminal	Specifies the destination for the End of Record event. When this event is asserted, the digitizer has completed sampling a record. Refer to the device specifications document for a list of valid destinations. Details
Synchronization:End of Record:Terminal Name	Returns the fully qualified name for the End of Record Event terminal. Details

Synchronization:End of Acquisition:Output Terminal	Specifies the destination for the End of Acquisition event. When this event is asserted, the digitizer has completed sampling all records. Refer to the device specifications document for a list of valid destinations. Details
Synchronization:End of Acquisition:Terminal Name	Returns the fully qualified name for the End of Acquisition Event terminal. Details
Synchronization:5 Volt Power:Output Terminal	Specifies the destination for the 5 Volt power signal. Refer to the device specifications document for a list of valid destinations. Details
Synchronization:Master Enable	Specifies whether the device is a master or a slave. Details
Acquisition:Binary Sample Width	Indicates the bit width of the binary data in the acquired waveform, which can help you determine which Binary Fetch to use. Details
Acquisition:Resolution	Indicates the actual resolution in bits of valid data (as opposed to padding bits) in the acquired waveform. Compare to the Binary Sample Width property. Valid Values: 8 to 32 Details
Acquisition:Acquisition Type	Specifies how the oscilloscope acquires data and fills the waveform record. Details
Acquisition:Sample Mode	Returns the sample mode the digitizer is currently using. Details
Acquisition:Advanced:Enable RIS in Auto Setup	Indicates whether the digitizer should use RIS sample rates when searching for a frequency in a utosetup. Details
Waveform Measurement:Other Channel	Specifies the second channel for two-channel measurements, such as Add Channels . If processing steps are registered with this channel, the processing happens before the waveform is used in a two-channel measurement. The default value is 0. Details

Waveform Measurement:Array Gain	Every element of an array is multiplied by this scalar value during the array gain measurement. The default value is 1.0. Details
Waveform Measurement:Array Offset	Every element of an array is added to this scalar value during the array offset measurement. The default value is 0.0. Details
Waveform Measurement:Hysteresis Percent	Digital hysteresis that is used in several of the scalar waveform measurements. This property specifies the percentage of the full-scale vertical range for the hysteresis window size. The default value is 2%. Details
Waveform Measurement>Last Acq. Histogram Size	Specifies the size (that is, the number of bins) in the last acquisition histogram. This histogram is used to determine several scalar measurements, most importantly voltage low and voltage high. The default value is 256. Details
Waveform Measurement:Voltage Histogram:Size	Specifies the number of bins in the running voltage histogram. The default value is 256. Details
Waveform Measurement:Voltage Histogram:Low Volts	Specifies the minimum voltage value in the running voltage histogram. The default value is -10.0. Details
Waveform Measurement:Voltage Histogram:High Volts	Specifies the maximum voltage value in the running voltage histogram. The default value is 10.0. Details
Waveform Measurement:Time Histogram:Size	Determines the multiple acquisition time histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with niScope Clear Waveform Measurement Stats . Details
Waveform Measurement:Time Histogram:Low Volts	Specifies the low voltage limit for the multi-acquisition time histogram. Only points in the waveform between the low and high voltage limits are included in the histogram. The default value is -10.0. Details

Waveform Measurement:Time Histogram:High Volts	Specifies the high voltage limit for the Multi-Acquisition time histogram. Only points in the waveform between the low and high voltage limits are included in the histogram. The default value is 10.0 V. Details
Waveform Measurement:Time Histogram:Low Time	Specifies the minimum time limit (in seconds) of the multi-acquisition time histogram, where the time is in seconds relative to the trigger position. Only points in the waveform between the low and high time limits are included in the histogram. The default value is -5.0e-4 . Details
Waveform Measurement:Time Histogram:High Time	Specifies the maximum time limit (in seconds) of the Multi-Acquisition time histogram, where the time is in seconds relative to the trigger position. Only points in the waveform between the low and high time limits are included in the histogram. The default value is 5.0e-4 . Details
Waveform Measurement:Interpolation:Polynomial Interpolation Order	Specifies the order of the polynomial used during the polynomial interpolation array measurement. For example, an order of 1 is linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid. The default value is 1 (linear interpolation). Details
Waveform Measurement:Interpolation:Sampling Factor	The new number of points for polynomial interpolation is the sampling factor times the input number of points. The default value is 0.0. Details
Waveform Measurement:Filter:Type	Specifies the type of digital filter. The default value is lowpass. Details
Waveform Measurement:Filter:FIR Taps	Specifies the number of taps for the finite impulse response filter. This value must be odd if the filter type is highpass or bandstop. Otherwise, the magnitude response goes to zero as the frequency goes to half the sampling rate. The default value is 25. Valid Values: >0 Details

Waveform Measurement:Filter:FIR Window	Specifies the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters. The default value is None (0). Details
Waveform Measurement:Filter:Width	Specifies the width (in Hz) of a bandpass or bandstop filter. The cutoff frequencies are the (center frequency property $\pm 0.5 \times$ filter width). The default value is 1.0e3. Details
Waveform Measurement:Filter:IIR Order	Specifies the order of the infinite impulse response filter. The default value is 2. Valid Values: >0 Details
Waveform Measurement:Filter:Cutoff Frequency	Specifies the cutoff frequency in hertz for filters of type lowpass and highpass. The cutoff frequency definition varies depending on the filter. The default value is 1.0e6 Hz. Details
Waveform Measurement:Filter:Center Frequency	The center frequency in hertz for filters of type bandpass and bandstop. The width of the filter is specified by Filter Width, where the cutoff frequencies are the center width. The default value is 1.0e6 Hz. Details
Waveform Measurement:Filter:Ripple	Specifies the amount of passband ripple (in dB) for Chebyshev filters. More ripple gives a sharper cutoff for a given filter order. The default value is 0.1. Valid Values: >0.0 Details
Waveform Measurement:Filter:Percent Waveform Transient	The percentage (0 - 100%) of the infinite impulse response (IIR) filtered waveform to eliminate from the beginning of the waveform. This action allows eliminating the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition. The default value is 20.0%. Valid Range: 0.0 - 100.0% Details

Waveform Measurement:Reference Levels:Channel Based Low Ref Level	Specifies the low reference level used in many scalar measurements. The default value is 10.0%. Details
Waveform Measurement:Reference Levels:Channel Based Mid Ref Level	Specifies the mid reference level used in many scalar measurements. The default value is 50%. Details
Waveform Measurement:Reference Levels:Channel Based High Ref Level	Specifies the high reference level used in many scalar measurements. The default value is 90%. Details
Waveform Measurement:Reference Levels:Units	Specifies the units for the waveform measurement reference levels. Details
Waveform Measurement:Reference Levels:Percentage Units Method	Specifies the method used to map percentage reference units to voltages. The default value is BaseTop. Details
Onboard Signal Processing:DDC:DDC Enabled	Enables/disables the digital downconverter (DDC) block of the digitizer. When the DDC block is disabled, all DDC-related properties are disabled and have no effect on the acquired signal. The default value is FALSE. Details
Onboard Signal Processing:DDC:Data Processing Mode	The way in which data is processed by the DDC block. The default value is Complex. Details
Onboard Signal Processing:DDC:Frequency Translation Enabled	Enables/disables frequency translating the data around the user-selected center frequency down to baseband. The default value is TRUE. Details
Onboard Signal Processing:DDC:Center Frequency	The frequency at which the DDC block frequency translates the input data. The default value is 10 MHz. Details
Onboard Signal Processing:DDC:Fetch Interleaved IQ Data	Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms. If enabled, the number of elements returned by scalar fetch types (such as 16-bit integer) is twice the requested number of samples. If disabled during DDC acquisitions in Compl

	ex mode, two noninterleaved arrays of data are returned per channel, per record. Details
Onboard Signal Processing:DDC:Q Source	Specifies the channel that is the input to the Q data stream of the DDC . Details
Onboard Signal Processing:DDC:Signal Adjustments:Frequency Translation:Frequency Translation Phase I	The I oscillator phase in degrees at the first point acquired. The default value is 0. Valid Values -360 to 360 Details
Onboard Signal Processing:DDC:Signal Adjustments:Frequency Translation:Frequency Translation Phase Q	The Q oscillator phase in degrees at the first point acquired. Use this property only when the Data Processing Mode property is set to Complex. The default value is 90. Valid Values -360 to 360 Details
Onboard Signal Processing:Equalization:Equalization Filter Enabled	Enables the onboard signal processing equalization FIR block, which is connected directly to the input signal. The equalization filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1. The default value is FALSE. Details
Onboard Signal Processing:Equalization:Equalization Num Coefficients	Returns the number of coefficients that the equalization FIR filter can accept. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1. Details
Onboard Signal Processing:Fractional Resample:Fractional Resample Enabled	Enables the onboard signal processing block that resamples the input waveform to the user desired sample rate. The default value is FALSE. Details

Onboard Signal Processing:OSP Overflow Error Reporting	Configures error reporting when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform. The default value is Warning. Details
Peer-to-Peer:P2P Enabled	Specifies whether the digitizer writes data to the peer-to-peer endpoint. This property is endpoint-based. Details
Peer-to-Peer:Channels to Stream	Specifies which channels will be written to a peer-to-peer endpoint. If multiple channels are specified, they will be interleaved by sample. This property is endpoint-based. The default value is 0. Details
Peer-to-Peer:Endpoint Size	Returns the size, in samples, of the peer-to-peer endpoint. This property is endpoint-based. Details
Peer-to-Peer:Samples Available In Endpoint	Returns the current number of samples available to stream from a peer-to-peer endpoint. This property is endpoint-based. Details
Peer-to-Peer:Most Samples Available in Endpoint	Returns the most number of samples available to stream from a peer-to-peer endpoint since the last time this property was read. This property is endpoint-based. Details
Peer-to-Peer:Samples Transferred	Returns the number of samples transferred through the peer-to-peer endpoint since the endpoint was last reset. This property is endpoint-based. Details
Peer-to-Peer:Endpoint Overflow	Returns TRUE if the peer-to-peer endpoint has overflowed. Reset the endpoint to clear the overflow condition. This property is endpoint-based. Details
Peer-to-Peer:FIFO Endpoint Count	Returns the number of FIFO-based peer-to-peer endpoints this device supports. Details

Peer-to-Peer:Onboard Memory Enabled	Specifies whether the digitizer writes data to on board memory when a peer-to-peer endpoint is enabled. Details
Peer-to-Peer:Stream Relative To	Determines which trigger peer-to-peer data is streamed relative to. The default value is Start Trigger. Details
Peer-to-Peer:Samples Transferred Per Record	Returns the number of samples transferred per record when you set the Stream Relative To property to Reference Trigger or Sync Trigger. Details
Peer-to-Peer:Manual:Manual Configuration Enabled	Enables and disables manual configuration of a peer-to-peer endpoint. These attributes cannot be used if an endpoint is being configured by NI-P2P, or a resource reservation error will result. This property is endpoint-based. Details
Peer-to-Peer:Manual:Configuration:Data Transfer Permission Address	Returns the address of a hardware register used to grant permission for the peer-to-peer endpoint to write data to another peer. The type of this address is determined by the Data Transfer Permission Address Type property. Permission is granted in bytes and the register is additive. This property is endpoint-based. Details
Peer-to-Peer:Manual:Configuration:Data Transfer Permission Address Type	Specifies the type of address returned to the user from the Data Transfer Permission Address property. This property is endpoint-based. Details
Peer-to-Peer:Manual:Configuration:Destination Window Address	Specifies the destination for data written by the peer-to-peer endpoint. The type of this address is specified by the Destination Window Address Type property. This property is endpoint-based. Details
Peer-to-Peer:Manual:Configuration:Destination Window Address Type	Specifies the type of the Destination Window Address property. This property is endpoint-based. Details
Peer-to-Peer:Manual:Configuration:Destination Window Size	Specifies the size, in bytes, of the destination window determined by the Destination Window

	Address and Destination Window Address Type properties. This property is endpoint-based. Details
Peer-to-Peer:Manual:Notification:Push Message On	Specifies the event to push the Message Push Value property to the Message Push Address property. Specifying Done will push the message when the acquisition has completed. This property is endpoint-based. Details
Peer-to-Peer:Manual:Notification:Message Push Address	Specifies the address to push the Message Push Value to on the event specified by the Push Message On property. This property is endpoint-based. Details
Peer-to-Peer:Manual:Notification:Message Push Address Type	Specifies the type of the Message Push Address property. This property is endpoint-based. Details
Peer-to-Peer:Manual:Notification:Message Push Value	Specifies the value to be pushed to the Message Push Address property on the event specified in the Push Message On property. This property is endpoint-based. Details
Device:Temperature	Returns the temperature of the device in degrees Celsius from the onboard sensor. Details
Device:Serial Number	Returns the serial number of the device. Details
Device:Accessory:Gain	Returns the calibration gain for the current device configuration. Details
Device:Accessory:Offset	Returns the calibration offset for the current device configuration. Details
Inherent IVI Attributes:User Options:Range Check	Specifies whether to validate property values and function parameters. If enabled, the instrument driver validates the parameter values that you pass to driver functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this property to FALSE to disable range checking and maximize performance. The default value is TRUE. Use

	<p>niScope Initialize with Options to override this value. Details</p>
Inherent IVI Attributes>User Options:Query Instrument Status	<p>Specifies whether the instrument driver queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this property to FALSE to disable status checking and maximize performance. The instrument driver can choose to ignore status checking for particular properties regardless of the setting of this property. The default value is TRUE. Use niScope Initialize with Options to override this value. Details</p>
Inherent IVI Attributes>User Options:Cache	<p>Specifies whether to cache the value of properties. When caching is enabled, the instrument driver keeps track of the current instrument settings and avoids sending redundant commands to the instrument. Thus, you can significantly increase execution speed. The instrument driver can choose always to cache or never to cache particular properties, regardless of the setting of this property. The default value is TRUE. Use niScope Initialize with Options to override this value. Details</p>
Inherent IVI Attributes>User Options:Simulate	<p>Specifies whether to simulate instrument driver I/O operations. The default value is FALSE. Use niScope Initialize with Options to override this value. Details</p>
Inherent IVI Attributes>User Options:Record Value Coercions	<p>Specifies whether the IVI engine keeps a list of the value coercions it makes for ViInt32 and DBL properties. The default value is FALSE. Use niScope Initialize with Options to override this value. Details</p>
Inherent IVI Attributes>User Options:Interchangeability Check	<p>Specifies whether to perform interchangeability checking and log interchangeability warnings when you call VIs. Interchangeability warnings indicate that using your application with a different</p>

	instrument might cause different behavior. Details
Inherent IVI Attributes:Driver Identification:Description	A string that contains the description of the instrument. Details
Inherent IVI Attributes:Driver Identification:Driver Prefix	A string that contains the prefix for the instrument driver. The name of each user-callable function in this driver starts with this prefix. Details
Inherent IVI Attributes:Driver Identification:Driver Vendor	A string that contains the name of the vendor that supplies this driver, for example, "National Instruments". Details
Inherent IVI Attributes:Driver Identification:Revision	The string that contains additional version information about this instrument driver. Details
Inherent IVI Attributes:Driver Identification:Class Specification Major Version	The major version number of the class specification with which this driver is compliant. Details
Inherent IVI Attributes:Driver Identification:Class Specification Minor Version	The minor version number of the class specification with which this driver is compliant. Details
Inherent IVI Attributes:Driver Capabilities:Supported Instrument Models	A string that contains a comma-separated list of the instrument model numbers supported by this driver. Details
Inherent IVI Attributes:Driver Capabilities:Class Group Capabilities	A string that contains a comma-separated list of class-extension groups that this driver implements. Details
Inherent IVI Attributes:Driver Capabilities:Channel Count	Indicates the number of channels that the specific instrument driver supports. For channel based properties, the IVI engine maintains a separate cache value for each channel. Details
Inherent IVI Attributes:Instrument Identification:Manufacturer	A string that contains the name of the instrument manufacturer, for example, "National Instruments". Details
Inherent IVI Attributes:Instrument Identification:Model	A string that contains the model number of the current instrument. Details

Inherent IVI Attributes:Instrument Identification:Firmware Revision	A string that contains the firmware revision information for the current instrument. Details
Inherent IVI Attributes:Advanced Session Information:Logical Name	A string that contains the logical name you specified when opening the current IVI session. Details
Inherent IVI Attributes:Advanced Session Information:Resource Descriptor	Indicates the resource descriptor the driver uses to identify the physical device. If you initialize the driver with a logical name, this property contains the resource descriptor that corresponds to the entry in the IVI Configuration utility. If you initialize the instrument driver with the resource descriptor, this property contains that value. Details
Device Specific:IF Digitizer:Fetch Interleaved Data	Specifies whether to interleave I and Q data retrieved from the IF digitizer in a single array or to retrieve two separate arrays, one for I data and another for Q data. Details
Device Specific:IF Digitizer:Device Number	Returns the device number of the IF digitizer associated with the current session. Details

Active Channel Property

Short Name: Active Channel

Property of [niScope](#)

Specifies the channel name used to access all subsequent properties in this instance of the property node. If the property you want to use is Channel Based, you need to first select this property and then pass the name of the specific channel. If the property you specify is not channel based, pass an empty string, or omit setting this property. The default value is "".

Remarks

The following table lists the characteristics of this property.

Datatype	
----------	---

Permissions	Write Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Vertical:Channel Enabled Property


Short Name: Channel Enabled

Property of [niScope](#)

Specifies whether the digitizer acquires a waveform for the channel.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Vertical
Channel-based	Yes
Resettable	Yes

Vertical:Vertical Range Property


Short Name: Vertical Range

Property of [niScope](#)

Specifies the absolute value of the input range for a channel. The units are volts. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Vertical

Channel-based	Yes
Resettable	Yes

Vertical:Vertical Offset Property


Short Name: Vertical Offset

Property of [niScope](#)

Specifies the location of the center of the range. The value is with respect to ground and is in volts. For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this property to 5.0 V. This property is not supported by all digitizers.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Vertical
Channel-based	Yes
Resettable	Yes

Vertical:Maximum Input Frequency Property

Short Name: Max Input Frequency

Property of [niScope](#)

Specifies the bandwidth of the channel in hertz. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.

Special Values:

(-1) Full bandwidth


(0) Device default

Related topics:

- [Bandwidth](#)
- [Probes and Their Effects](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Chan Characteristics
Channel-based	Yes
Resettable	Yes

Vertical:Vertical Coupling Property

Short Name: Vertical Coupling

Property of [niScope](#)

Specifies how the digitizer couples the input signal for the channel. When you change input coupling, the input stage takes a finite amount of time to settle.


Related topics:

- [Input Coupling](#)

AC (0)	AC coupled
DC (1)	DC coupled
Ground (2)	Ground coupled

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Vertical
Channel-based	Yes
Resettable	No

Vertical:Input Impedance Property

Short Name: Input Impedance

Property of [niScope](#)

Specifies the input impedance for the channel in ohms.

Defined Values

50 Ohm


1 M Ohm

Related topics:

- [Impedance and Impedance Matching](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Chan Characteristics
Channel-based	Yes
Resettable	No

Vertical:Probe Attenuation Property

Short Name: Probe Attenuation

Property of [niScope](#)


Specifies the probe attenuation for the input channel. For example, for a 10:1 probe, you would set this property to 10.0.

Related topics:

- [Probes and Their Effects](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Vertical
Channel-based	Yes
Resettable	Yes

Vertical:Channel Terminal Configuration Property

Short Name: Terminal Configuration

Property of [niScope](#)

Specifies how the digitizer configures the channel terminal.


Related topics:

- [NI 5922 Channel Terminal Configuration](#)

Single Ended (0)	Single-ended channel terminal configuration.
Unbalanced Differential (1)	Unbalanced differential channel terminal configuration.
Differential (2)	Differential channel terminal configuration.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:Digital Gain Property

Short Name: Digital Gain

Property of [niScope](#)

Applies gain to the specified channel in hardware before any onboard signal processing occurs. The default value is 1.

The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset

Units: Unitless

Valid Values: -1.5 to 1.5



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Related topics:

- [NI 5622 Onboard Signal Processing \(OSP\)](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:Digital Offset Property

Short Name: Digital Offset

Property of [niScope](#)

Applies offset to the specified channel in hardware before any onboard signal processing occurs. The default value is 0.


Units: Volts

Valid Values

$\pm(\text{Vertical Range} \times 0.4)$

The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset


 **Note** This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Related topics:

- [NI 5622 Onboard Signal Processing \(OSP\)](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:Bandpass Filter Enabled Property

Short Name: Bandpass Filter Enabled

Property of [niScope](#)


Enables the bandpass filter on the specified channel. For the NI PXIe-5622, set the value to TRUE to enable the IF filtered path 50MHz bandpass filter centered at 187MHz. The default value is FALSE.

Related topics:

- [Bandwidth](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:Dither Enabled Property


Short Name: Dither Enabled

Property of [niScope](#)

Enables or disables the analog dither on the device. Using dither can improve the spectral performance of the device by reducing the effects of quantization. However, adding dither increases the power level to the ADC, so you may need to either decrease the signal level or increase the vertical range. The default value is FALSE.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:Flex FIR Antialias Filter Type Property

Short Name: Flex FIR Antialias Filter Type

Property of [niScope](#)

The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter. Use this property to select from several types of filters to achieve desired filtering characteristics. For most applications, the default value of this property is recommended. The other available filters are useful for optimizing settling time measurements of step responses. The default value is 48 Tap Standard.



Note Settling time values refer to the FIR filter only and do not take into account settling time caused by the analog front end. Refer to the **NI PXI-5922 Specifications** for combined digital and analog settling times.


Related topics:

- [Aliasing](#)
- [FIR Filters](#)

48 Tap Standard (0)	48 Tap Standard filter is optimized for alias protection and frequency-domain flatness.
48 Tap Hanning (1)	48 Tap Hanning filter is optimized for the lowest possible bandwidth for a 48 tap filter and maximizes the SNR.
16 Tap Hanning (2)	16 Tap Hanning is optimized for the lowest possible bandwidth for a 16 tap filter and maximizes the SNR.
8 Tap Hanning (3)	8 Tap Hanning filter is optimized for the lowest possible bandwidth for a 8 tap filter and maximizes the SNR.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Vertical:Advanced:High Pass Filter Frequency Property

Short Name: High Pass Filter Frequency

Property of [niScope](#)

Specifies the frequency for the highpass filter in Hz. The device uses one of the valid values listed below. If an invalid value is specified, no coercion occurs. The default value is 0.

(PXIe-5164) Valid Values:

- 0
- 90
- 450

Related topics:

- [Digital Filtering](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	No
Resettable	No

Vertical:Advanced:Interleaving Offset Correction Enabled Property

Short Name: Interleaving Offset Correction Enabled

Property of [niScope](#)

Enables the interleaving offset correction on the specified channel. The default value is TRUE.



Note If disabled, warranted specifications are not guaranteed.

Related topics:

- [Timed Interleaved Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:Min Sample Rate Property

Short Name: Min Sample Rate

Property of [niScope](#)

Specifies the sampling rate (in Samples/second) for the acquisition. This attribute is invalid when the device is configured to use an external sample clock timebase.


When a DDC is enabled, this attribute specifies the IQ rate. When both the [Time Per Record](#) Property and the Min Sample Rate Property are set, the attribute that was set first is ignored.

Valid Values: The combination of sampling rate and minimum record length must allow the digitizer to sample at a valid sampling rate for the acquisition type specified in the [niScope Configure Acquisition](#) VI and not require more memory than the onboard memory module allows.

- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Horizontal Timing
Channel-based	No
Resettable	Yes

Horizontal:Min Number of Points Property

Short Name: Min Number of Points

Property of [niScope](#)

Specifies the minimum number of points you require in the waveform record for each channel.

NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. The [Actual Record Length](#) property returns the actual record length.

Related topics:

- [Record Length](#)
- [Sample Rate](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Horizontal Timing
Channel-based	No
Resettable	Yes

Horizontal: Number of Records Property

Short Name: Number of Records

Property of [niScope](#)

Specify the number of records to acquire.

Related topics:

- [Making Multiple-Record Acquisitions](#)
- [Fetching Multiple-Record Acquisitions](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Horizontal Timing
Channel-based	No
Resettable	No

Horizontal: Reference Position Property

Short Name: Reference Position


Property of [niScope](#)

Specifies the position of the Reference Event in the waveform record as a percentage of the record.

When the digitizer detects a trigger, it waits the length of time the [Trigger Delay](#) property specifies. The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Horizontal Timing
Channel-based	No
Resettable	Yes

Horizontal:Actual Sample Rate Property

Short Name: Actual Sample Rate

Property of [niScope](#)

Returns the actual sample rate used for the acquisition.


Units: hertz (Samples / Second)

Related topics:

- [Sample Clock](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	niScope Sample Rate

Channel-based	No
Resettable	No

Horizontal:Actual Record Length Property

Short Name: Actual Record Length

Property of [niScope](#)

Returns the actual number of points the digitizer acquires for each channel. The value is equal to or greater than the value you specify in the [niScope Configure Horizontal Timing VI](#).

Valid Values: 1 to the maximum memory size

- [Record Length](#)
- [Coercions of Horizontal Parameters](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	niScope Actual Record Length
Channel-based	No
Resettable	No

Horizontal:Enable Time Interleaved Sampling Property

Short Name: Enable TIS

Property of [niScope](#)

Extends the maximum sample rate on the specified Active Channel for some devices that support Time Interleaved Sampling (TIS). TIS enables the device to use multiple ADCs to sample the same waveform at a higher effective real-time rate. NI


5152/5153/5154 devices fully support Read/Write ability for this property. For other devices that use TIS mode, such as the NI 5185/5186, this property is Read Only.

Related topics:

- [Time Interleaved Sampling](#)
- [Configuring the Horizontal Settings](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Horizontal:Enforce Realtime Property

Short Name: Enforce Realtime

Property of [niScope](#)


Indicates whether the digitizer enforces real-time measurements or allows equivalent-time measurements.

Related topics:

- [Sampling Methods](#)
- [Real-Time Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Horizontal Timing
Channel-based	No

Horizontal:Enable Records > Memory Property

Short Name: Enable Records > Memory

Property of [niScope](#)

Allows you to acquire more records than fit in onboard memory.

TRUE—Enables NI-SCOPE to fetch more records than fit in memory

FALSE—Disables NI-SCOPE from fetching more records than fit in memory



Note The property can be used only in digitizers that support continuous acquisition. Refer to [Features Supported by Device](#) to find out if your digitizer supports continuous acquisition.

Related topics:

- [Time Interleaved Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:RIS Num Avg Property

Short Name: RIS Num Avg

Property of [niScope](#)

Specifies the number of averages in each RIS bin.

Averaging is useful in RIS because the trigger times are not evenly spaced, so adjacent points in the reconstructed waveform cannot be accurately spaced. By

averaging, the errors in both time and voltage are smoothed, minimizing the noise in the reconstructed waveform.

Valid Values: Greater than or equal to 0

Related topics:

- [Sampling Methods](#)
- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:RIS Method Property

Short Name: RIS Method

Property of [niScope](#)


Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the [Max Realtime Sample Rate](#).

Exact Num Avg. (1)	Acquires exactly the specified number of records for each bin in the RIS acquisition.
Min Num Avg. (2)	Each RIS sample is the average of a least a minimum number of randomly distributed points.
Incomplete (3)	If RIS does not complete in the allotted fetch time, the Fetch VI should abort and return the incomplete data. Any missing samples appear as NaN when fetching scaled data or zero when fetching binary data. A warning with a positive error code is returned from the Fetch VI if the RIS acquisition is incomplete.

	sition did not finish. The acquisition is aborted when data is returned.
Limit Bin Width (5)	Each RIS sample is the average of Min Num Avg points distributed close to the sample period boundaries (within 200 ps). Points falling between sample periods are ignored.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:Maximum Real Time Sample Rate Property

Short Name: Maximum Real Time Sample Rate

Property of [niScope](#)


Returns the maximum real-time sample rate in hertz.

Related topics:

- [Sampling Methods](#)
- [Real-Time Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A

Channel-based	No
Resettable	No

Horizontal:Maximum RIS Rate Property

Short Name: Maximum RIS Rate

Property of [niScope](#)


Returns the maximum RIS sampling rate in hertz.

Related topics:

- [Sampling Methods](#)
- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:Memory Size Property


Short Name: Memory Size

Property of [niScope](#)

Returns the total combined amount of onboard memory for all channels in bytes.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A

Channel-based	No
Resettable	No

Horizontal:Advanced:Enable TDC Property

Short Name: Enable TDC

Property of [niScope](#)


Specifies that the digitizer should record the trigger position precisely using time-digital conversion (TDC).

Related topics:

- [TDC](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Horizontal:Advanced:Time Per Record Property

Short Name: Time Per Record

Property of [niScope](#)

Specifies the length of time (in seconds) that corresponds to the record length. This attribute is invalid when the device is configured to use an external sample clock timebase. This attribute is also invalid when a DDC is enabled. When both the Time Per Record Property and the [Min Sample Rate](#) Property are set, the attribute that was set first is ignored.


Related topics:

- [Record Length](#)

- [Sample Rate](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	Yes

Horizontal:Advanced:Acquisition Start Time Property

Short Name: Acq Start Time


Property of [niScope](#)

Specifies the length of time (in seconds) from the trigger event to the first point in the waveform record.

If the value is positive, the first point in the waveform record occurs after the trigger event (same as specifying a trigger delay). If the value is negative, the first point in the waveform record occurs before the trigger event (same as specifying Reference Position).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	Yes

Triggering:Trigger Type Property

Short Name: Trigger Type

Property of [niScope](#)

Specifies the type of trigger to use.

Edge (1)	Specifies an edge trigger.
Width (2)	Specifies a width trigger.
Runt (3)	Specifies a runt trigger.
Glitch (4)	Specifies a glitch trigger.
Video (5)	Specifies a video trigger.
Immediate (6)	Specifies an immediate trigger.
Hysteresis (1001)	Specifies a hysteresis trigger.
Digital (1002)	Specifies a digital trigger.
Window (1003)	Specifies a window trigger.
Software (1004)	Specifies a software trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering:Trigger Source Property

Short Name: Trigger Source


Property of `niScope`

Specifies the source the digitizer monitors for the trigger event. The value must be selected from one of the following valid values.

0..n	n is the number of channels on the device
VAL_EXTERNAL	External TRIG input
VAL_IMMEDIATE	Triggers immediately
VAL_RTSI_0	RTSI 0
VAL_RTSI_1	RTSI 1
VAL_RTSI_2	RTSI 2
VAL_RTSI_3	RTSI 3
VAL_RTSI_4	RTSI 4
VAL_RTSI_5	RTSI 5
VAL_RTSI_6	RTSI 6
VAL_PFI_0	PFI 0
VAL_PFI_1	PFI 1
VAL_PFI_2	PFI 2
VAL_PFI_3	PFI 3
VAL_PFI_4	PFI 4
VAL_PFI_5	PFI 5
VAL_PFI_6	PFI 6
VAL_PFI_7	PFI 7
VAL_PXI_STAR	PXI Star trigger
VAL_SW_TRIG_FUNC	Waits for niScope Send Software Trigger Edge

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Digital (Digital Edge Ref Trigger) , niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Glitch (Glitch Trigger) , niScope Configure Trigger

	Hysteresis (Analog Hysteresis Ref Trigger) , niScope Configure Trigger Runt (Runt Trigger) , niScope Configure Trigger Video (Video Ref Trigger) , niScope Configure Trigger Width (Width Trigger) , niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Output Terminal Property

Short Name: RefTrig.OutputTerm

Property of [niScope](#)

Specifies the destination to export the Reference (Stop) Trigger Refer to the device specifications document for a list of valid destinations.

Defined Values

VAL_EXTERNAL

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5


VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resetable	No

Triggering:Terminal Name Property

Short Name: Reference Trigger Terminal Name


Property of [niScope](#)

Returns the fully qualified name for the Reference Trigger terminal.

You can use this terminal as the source for another trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Triggering:Trigger Level Property

Short Name: Trigger Level

Property of [niScope](#)

Specifies the voltage threshold for the trigger. The units are volts.

This property affects instrument behavior only when the [Trigger Type](#) is set to Edge, Hysteresis, or Window.


The values of the range and offset parameters in [niScope Configure Vertical](#) determine the valid range for the trigger level on the channel you use as the trigger source. The value you pass for this parameter must meet the following conditions:

Trigger Level \leq Vertical Range/2 + Vertical Offset

Trigger Level \geq (-Vertical Range/2) + Vertical Offset

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Glitch (Glitch Trigger) , niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger) , niScope Configure Trigger Width (Width Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Modifier Property

Short Name: Trigger Modifier

Property of [niScope](#)

Configures the device to automatically complete an acquisition if a trigger has not been received.




Note When Auto Trigger is selected, you may need to modify the timeout on calls to fetch data to ensure that the acquisition does not time out before the auto trigger time elapses.

None (1)	Normal triggering.
Auto Trigger (2)	Software will trigger an acquisition automatically if no trigger arrives after a certain amount of time.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering:Auto Triggered Property

Short Name: Auto Triggered

Property of [niScope](#)


Specifies whether the acquisition was triggered automatically. Auto triggering occurs if the Trigger Modifier property is set to Auto Trigger and no trigger has been received for a certain amount of time.

Related topics:

- [Trigger Types](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering: Trigger Hysteresis Property

Short Name: Trigger Hysteresis

Property of [niScope](#)

Specifies the size of the hysteresis window, in volts, on either side of the trigger level.

The digitizer triggers when the trigger signal passes through the threshold you specify with the Trigger Level parameter, has the slope you specify with the Trigger Slope parameter, and passes through the hysteresis window that you specify with this property.


Default Value: 0.05

Valid Values

Positive trigger slope	$0 \leq \mathbf{Hysteresis} \leq \mathbf{Trigger\ Level} + (\mathbf{Vertical\ Range} / 2) - \mathbf{Vertical\ Offset}$
Negative trigger slope	$0 \leq \mathbf{Hysteresis} \leq (\mathbf{Vertical\ Range} / 2) + \mathbf{Vertical\ Offset} - \mathbf{Trigger\ Level}$

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Delay Property

Short Name: Trigger Delay


Property of [niScope](#)

Specifies the trigger delay time in seconds.

The trigger delay time is the length of time the digitizer waits after it receives the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Digital (Digital Edge Ref Trigger) , niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Glitch (Glitch Trigger) , niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger) , niScope Configure Trigger Runt (Runt Trigger) , niScope Configure Trigger Software (Software Ref Trigger) , niScope Configure Trigger Video (Video Ref Trigger) , niScope Configure Trigger Width (Width Trigger) , niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resettable	Yes

Triggering: Trigger Holdoff Property

Short Name: Trigger Holdoff


Property of [niScope](#)

Specifies the length of time the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger. The units are seconds.

This property affects instrument operation only when the digitizer requires multiple acquisitions to build a complete waveform. The digitizer requires multiple waveform acquisitions when it uses equivalent-time sampling or when the digitizer is configured for a multirecord acquisition through a call to [niScope Configure Horizontal Timing](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Digital (Digital Edge Ref Trigger) , niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Glitch (Glitch Trigger) , niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger) , niScope Configure Trigger Runt (Runt Trigger) , niScope Configure Trigger Software (Software Ref Trigger) , niScope Configure Trigger Video (Video Ref Trigger) , niScope Configure Trigger Width (Width Trigger) , niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resetable	No

Triggering: Trigger Slope Property

Short Name: Trigger Slope

Property of [niScope](#)


Specifies whether a rising or a falling edge triggers the digitizer.

This property affects instrument operation only when the [Trigger Type](#) property is set to edge, hysteresis, window, or video.

Positive (1)	Specifies a rising edge (positive slope).
Negative (0)	Specifies a falling edge (negative slope).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	niScope Configure Trigger Digital (Digital Edge Ref Trigger) , niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Coupling Property

Short Name: Trigger Coupling

Property of [niScope](#)

Specifies how the digitizer couples the trigger source.

This property affects instrument operation only when the [Trigger Type](#) property is set to Edge, Hysteresis, Window, or Video. If the trigger source is an input channel, the coupling of that channel is used for the trigger.

AC (0)	AC coupled
DC (1)	DC coupled
HF Reject (3)	HF Reject filter.
LF Reject (4)	LF Reject filter.
AC Plus HF Reject (1001)	AC Plus HF Reject filter.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Edge (Analog Edge Ref Trigger) , niScope Configure Trigger Glitch (Glitch Trigger) , niScope Configure Trigger Hysteresis (Analog Hysteresis Ref Trigger) , niScope Configure Trigger Runt (Runt Trigger) , niScope Configure Trigger Video (Video Ref Trigger) , niScope Configure Trigger Width (Width Trigger) , niScope Con

	figure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resetttable	No

Triggering: Trigger Impedance Property

Short Name: Trigger Impedance

Property of [niScope](#)

Sets the impedance for the external trigger input.

Supported Devices

- [PXIe-5160](#) (2 CH only)
- [PXIe-5162](#) (2 CH only)


Defined Values

1 M Ohm

50 Ohm

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetttable	No

Triggering: Start To Ref Trigger Holdoff Property

Short Name: Start To Ref Trigger Holdoff


Property of [niScope](#)

Pass the length of time (in seconds) you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a reference (stop) trigger.

Valid Values: 0.0 - 171.8

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering: End of Record to Advance Trigger Holdoff Property

Short Name: End of Record to Advance Trigger Holdoff


Property of [niScope](#)

End of Record to Advance Trigger Holdoff is the length of time (in seconds) that a device waits between the completion of one record and the acquisition of pre-trigger samples for the next record. During this time, the acquisition engine state delays the transition to the Wait for Advance Trigger state, and will not store samples in onboard memory, accept an Advance Trigger, or trigger on the input signal..

Supported Devices: NI 5185/5186

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A

Channel-based	No
Resettable	No

Triggering: Trigger Window: Window Mode Property

Short Name: Trigger Window Mode

Property of [niScope](#)

Specifies whether to trigger when the signal enters or leaves the window specified by the [Trigger Window Low Level](#) property or the [Trigger Window High Level](#) property.

Related topics:

- [Window Triggers](#)
- [Trigger Parameters](#)

Entering (0)	Trigger occurs when a signal enters a window.
Leaving (1)	Trigger occurs when a signal leaves a window.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Window: Low Level Property

Short Name: Trigger Window Low Level

Property of [niScope](#)

Pass the lower voltage threshold you want the digitizer to use for window triggering.

The digitizer triggers when the trigger signal enters or leaves the window you specify with this property and the [Trigger Window High Level](#) property.

The values of the [Vertical Range](#) property and the [Vertical Offset](#) property determine the valid range for this property on the channel you specify with the [Trigger Source](#) property.

The value you pass for this parameter must meet the following conditions:

Low Trigger Level \leq Vertical Range/2 + Vertical Offset

Low Trigger Level \geq (-Vertical Range/2) + Vertical Offset


Low Trigger Level < High Trigger Level

Related topics:

- [Window Triggers](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Window: High Level Property

Short Name: Trigger Window High Level

Property of [niScope](#)

Pass the upper voltage threshold you want the digitizer to use for window triggering.

The digitizer triggers when the trigger signal enters or leaves the window you specify with the Trigger Window Low Level property and this property.

The values of the [Vertical Range](#) property and the [Vertical Offset](#) property determine the valid range for the `Trigger Window Low Level` property on the channel you specify with the [Trigger Source](#) property.

The value you pass for this parameter must meet the following conditions:

High Trigger Level \leq Vertical Range/2 + Vertical Offset

High Trigger Level \geq (-Vertical Range/2) + Vertical Offset


High Trigger Level $>$ Low Trigger Level

Related topics:

- [Window Triggers](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Window (Analog Window Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Video: Signal Format Property

Short Name: Video Signal Format

Property of [niScope](#)

Specifies the video signal format to use.

M-NTSC (1)	Specifies M-NTSC signal format.
B/G-PAL (2)	Specifies BG/PAL signal format.
SECAM (3)	Specifies SECAM signal format.
M-PAL (1001)	Specifies M-PAL signal format.

480i/59.94 fps (1010)	Specifies 480i/59.94 fps signal format.
480i/60 fps (1011)	Specifies 480i/60 fps signal format.
480p/59.94 Fps (1015)	Specifies 480p/59.94 Fps signal format.
480p/60 Fps (1016)	Specifies 480p/60 Fps signal format.
576i/50 fps (1020)	Specifies 576i/50 fps signal format.
576p/50 Fps (1025)	Specifies 576p/50 Fps signal format.
720p/50 Fps (1031)	Specifies 720p/50 Fps signal format.
720p/59.94 Fps (1032)	Specifies 720p/59.94 Fps signal format.
720p/60 Fps (1033)	Specifies 720p/60 Fps signal format.
1080i/50 fps (1040)	Specifies 1080i/50 fps signal format.
1080i/59.94 fps (1041)	Specifies 1080i/59.94 fps signal format.
1080i/60 fps (1042)	Specifies 1080i/60 fps signal format.
1080p/24 Fps (1045)	Specifies 1080p/24 Fps signal format.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Video (Video Ref Trigger)
Channel-based	Yes
Resettable	No

Triggering: Trigger Video: Line Number Property

Short Name: Video Line Number

Property of [niScope](#)

Specifies the line number to trigger on.

This property is only used if the video trigger [Event](#) property is set as Line Number. Valid values depend on the video signal format selected.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Video (Video Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Video: Polarity Property

Short Name: Video Polarity

Property of [niScope](#)

Specifies whether the video signal is positive or negative.

Positive (1)	Specifies that the video signal has positive polarity.
Negative (2)	Specifies that the video signal has negative polarity.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Video (Video Ref Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Video:Event Property

Short Name: Video Trigger Event

Property of [niScope](#)

Specifies the event to trigger on.

Field 1 (1)	Trigger on field 1 of the signal.
Field 2 (2)	Trigger on field 2 of the signal.
Any Field (3)	Trigger on any field of the signal.
Any Line (4)	Trigger on the first line acquired.
Line Number (5)	Trigger on a specific line of a video signal. Valid values vary depending on the signal format.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Video (Video Ref Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Video:Enable DC Restore Property

Short Name: Enable DC Restore

Property of [niScope](#)

Restores the video-triggered data retrieved by the digitizer to the video signal's zero reference point. The default value is FALSE.

Remarks

The following table lists the characteristics of this property.

Datatype	TF1
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Video (Video Ref Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Glitch: Glitch Condition Property

Short Name: Glitch Condition

Property of [niScope](#)

Specifies whether the oscilloscope triggers on pulses of duration less than or greater than the specified [Glitch Width](#).

Default Value: Greater Than

Less Than (1)	The oscilloscope triggers on pulses with a duration shorter than the specified glitch width.
Greater Than (2)	The oscilloscope triggers on pulses with a duration longer than the specified glitch width.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Glitch (Glitch Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Glitch:Glitch Width Property

Short Name: Glitch Width

Property of [niScope](#)


Specifies the glitch duration, in seconds.

The oscilloscope triggers when it detects a pulse of duration either less than or greater than this value depending on the value of the [Glitch Condition](#) property.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Glitch (Glitch Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Glitch:Glitch Polarity Property

Short Name: Glitch Polarity

Property of [niScope](#)

Specifies the polarity of pulses that trigger the oscilloscope for glitch triggering.

Default Value: Positive

Positive (1)	The oscilloscope triggers on pulses of positive polarity relative to the trigger threshold.
Negative (2)	The oscilloscope triggers on pulses of negative polarity relative to the trigger threshold.
Either (3)	The oscilloscope triggers on pulses of either positive or negative polarity.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Glitch (Glitch Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Width: Width Condition Property

Short Name: Width Condition

Property of [niScope](#)

Specifies whether the oscilloscope triggers on pulses within or outside the duration range bounded by the [Width Low Threshold](#) and [Width High Threshold](#) properties.

Default Value: Within

Within (1)	The oscilloscope triggers on pulses with a duration within the range bounded by Width Low Threshold and Width High Threshold .
Outside (2)	The oscilloscope triggers on pulses with a duration not within the range bounded by Width Low Threshold and Width High Threshold .

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Width (Width Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Width:Width Low Threshold Property

Short Name: Width Low Threshold

Property of [niScope](#)


Specifies the low width threshold, in seconds.

This property sets the lower bound on the duration range that triggers the oscilloscope. The [Width Condition](#) property determines how the oscilloscope triggers in relation to the width thresholds.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Width (Width Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Width:Width High Threshold Property

Short Name: Width High Threshold

Property of [niScope](#)


Specifies the high width threshold, in seconds.

This property sets the upper bound on the duration range that triggers the oscilloscope. The [Width Condition](#) property determines how the oscilloscope triggers in relation to the width thresholds.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Width (Width Trigger)
Channel-based	No
Resettable	No

Triggering: Trigger Width: Width Polarity Property

Short Name: Width Polarity

Property of [niScope](#)


Specifies the polarity of pulses that trigger the oscilloscope for width triggering.

Default Value: Positive

Positive (1)	The oscilloscope triggers on pulses of positive polarity relative to the trigger threshold.
Negative (2)	The oscilloscope triggers on pulses of negative polarity relative to the trigger threshold.
Either (3)	The oscilloscope triggers on pulses of either positive or negative polarity.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Width (Width Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Runt:Runt Low Threshold Property

Short Name: Runt Low Threshold

Property of [niScope](#)


Specifies the lower of two thresholds, in volts, that bound the vertical range to examine for runt pulses.

The runt threshold that causes the oscilloscope to trigger depends on the runt polarity you select. Refer to the [Runt Polarity](#) property for more information.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Runt (Runt Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Runt:Runt High Threshold Property

Short Name: Runt High Threshold

Property of [niScope](#)


Specifies the higher of two thresholds, in volts, that bound the vertical range to examine for runt pulses.

The runt threshold that causes the oscilloscope to trigger depends on the runt polarity you select. Refer to the [Runt Polarity](#) property for more information.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Runt (Runt Trigger)
Channel-based	No
Resettable	No

Triggering:Trigger Runt:Runt Polarity Property

Short Name: Runt Polarity

Property of [niScope](#)

Specifies the polarity of pulses that trigger the oscilloscope for runt triggering.

This property determines how the oscilloscope triggers relative to the runt thresholds you set.

When set to **Positive**, the oscilloscope triggers when the following conditions are met:

- The leading edge of a pulse crosses the [Runt Low Threshold](#) in a positive direction;
- The trailing edge of the pulse crosses the [Runt Low Threshold](#) in a negative direction; and
- No portion of the pulse crosses the [Runt High Threshold](#).

When set to **Negative**, the oscilloscope triggers when the following conditions are met:

- The leading edge of a pulse crosses the [Runt High Threshold](#) in a negative direction;
- The trailing edge of the pulse crosses the [Runt High Threshold](#) in a positive direction; and

- No portion of the pulse crosses the [Runt Low Threshold](#).

When set to Either, the oscilloscope triggers in either case.

Default Value: Positive

Positive (1)	The oscilloscope triggers on pulses of positive polarity relative to Runt Low Threshold that do not cross Runt High Threshold .
Negative (2)	The oscilloscope triggers on pulses of negative polarity relative to the Runt High Threshold that do not cross Runt Low Threshold .
Either (3)	The oscilloscope triggers on pulses of either positive or negative polarity.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Trigger Runt (Runt Trigger)
Channel-based	No
Resetttable	No

Triggering:Trigger Runt:Runt Time Condition Property

Short Name: Runt Time Condition

Property of [niScope](#)

Specifies whether runt triggers are time qualified, and if so, how the oscilloscope triggers in relation to the duration range bounded by the [Runt Time Low Limit](#) and [Runt Time High Limit](#) properties.

Default Value: None

None (0)	Time qualification is disabled. The oscilloscope triggers on runt pulses based solely on the voltage level of the pulses.
Within (1)	The oscilloscope triggers on pulses that, in addition to meeting runt voltage criteria, have a duration within the range bounded by Runt Time Low Limit and Runt Time High Limit .
Outside (2)	The oscilloscope triggers on pulses that, in addition to meeting runt voltage criteria, have a duration not within the range bounded by Runt Time Low Limit and Runt Time High Limit .

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering: Trigger Runt: Runt Time Low Limit Property

Short Name: Runt Time Low Limit

Property of [niScope](#)


Specifies, in seconds, the low runt threshold time.

This property sets the lower bound on the duration of runt pulses that may trigger the oscilloscope. The [Runt Time Condition](#) property determines how the oscilloscope triggers in relation to the runt time limits.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering:Trigger Runt:Runt Time High Limit Property

Short Name: Runt Time High Limit

Property of [niScope](#)


Specifies, in seconds, the high runt threshold time.

This property sets the upper bound on the duration of runt pulses that may trigger the oscilloscope. The [Runt Time Condition](#) property determines how the oscilloscope triggers in relation to the runt time limits.

Default Value: 0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering:Onboard Signal Processing:Ref Trigger Detection Location Property

Short Name: Ref Trigger Detector Location

Property of [niScope](#)

Specifies which reference trigger detection circuitry to use on the device.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Analog Detection Circuit (0)	(Default) Uses the hardware analog circuitry to implement the reference trigger. This option detects trigger conditions by analyzing the unprocessed analog signal.
DDC Output (1)	Uses the onboard signal processing logic to implement the reference trigger. This option detects trigger conditions by analyzing the processed digital signal.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Triggering:Onboard Signal Processing:Ref Trigger Min Quiet Time Property

Short Name: Ref Trigger Min Quiet Time

Property of [niScope](#)

Specifies the amount of time (in seconds) the trigger circuit must not detect a signal above the [trigger level](#) (or below the trigger level if the trigger slope is negative) before the trigger is armed. This property is useful for triggering at the beginning of signal bursts instead of in the middle of signal bursts. The default value is 0.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Valid Values

Any value greater than or equal to 0.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Device:CableSense:Voltage Property

Short Name: CableSense Voltage

Property of [niScope](#)

Returns the voltage of the CableSense signal that is written to the EEPROM of the oscilloscope during factory calibration.


Supported Devices

- [PXIe-5110](#)
- [PXIe-5111](#)
- [PXIe-5113](#)
- [PXIe-5160](#)

- [PXIe-5162](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Device:CableSense:Mode Property

Short Name: CableSense Mode

Property of [niScope](#)

Specifies whether and how the oscilloscope is configured to generate a CableSense signal on the specified channels when the [niScope CableSense Signal Start VI](#) is called.



Note The input impedance of the channel(s) to convey the CableSense signal must be set to 50 ohms.

Device-Specific Behavior

PXIe-5160 PXIe-5162	<ul style="list-style-type: none"> ▪ The value of this property must be identical across all channels whose input impedance is set to 50 ohms. ▪ If this property is set to a value other than Disabled for any channel(s), the input impedance of all channels for which this property is set to Disabled must be set to 1 M Ohm.
------------------------	--

Supported Devices

- [PXIe-5110](#)
- [PXIe-5111](#)

- [PXIe-5113](#)
- [PXIe-5160](#)
- [PXIe-5162](#)

Disabled (0)	The oscilloscope is not configured to emit a CableSense signal.
On Demand (1)	The oscilloscope is configured to emit a single CableSense pulse.

Remarks

The following table lists the characteristics of this property.


Datatype	132
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Device:FPGA Bitfile Path Property

Short Name: FPGA Bitfile Path

Property of [niScope](#)

Gets the absolute file path to the bitfile loaded on the FPGA.

 Note Gets the absolute file path to the bitfile loaded on the FPGA.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Device:Temperature Property

Short Name: Device Temperature

Property of [niScope](#)


Returns the temperature of the device in degrees Celsius from the onboard sensor.

Related topics:

- [Thermal Shutdown](#)
- [PXI/PXIe Chassis Cooling](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Device:Serial Number Property


Short Name: Serial Number

Property of [niScope](#)

Returns the serial number of the device.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No

Device:Accessory:Gain Property

Resetable

No

Short Name: Accessory Gain

Property of [niScope](#)

Returns the calibration gain for the current device configuration.



Note This property is only supported by the NI PXI-5900 differential amplifier.

Related topics:

- [NI 5122/5124/5142 Calibration](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Resetable	No

Device:Accessory:Offset Property

Short Name: Accessory Offset

Property of [niScope](#)

Returns the calibration offset for the current device configuration.




Note This property is supported only by the NI PXI-5900 differential amplifier.

Related topics:

- [NI 5122/5124/5142 Calibration](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Acquisition:Enabled Channels Property

Short Name: Enabled Channels

Property of [niScope](#)

Returns a comma-separated list of the channels enabled for the session in ascending order.

Property-Specific Syntax


- If no channels are enabled, this property returns an empty string, "".
- If all channels are enabled, this property enumerates all of the channels.



Note Because this property returns channels in ascending order, but the order in which you specify channels for the channels input to NI-SCOPE VIs is important, the value of this property may not necessarily reflect the order in which NI-SCOPE performs certain actions. Refer to [Channel String Syntax](#) for more information on the effects of channel order in NI-SCOPE.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Acquisition:Binary Sample Width Property

Short Name: Binary Sample Width

Property of [niScope](#)

Indicates the bit width of the binary data in the acquired waveform, which can help you determine which Binary Fetch to use.

To configure the device to store samples with a lower resolution than the native, set this property to the desired binary width. This configuration can be useful for streaming at faster speeds, but at the cost of resolution. The least significant bits are lost with this configuration. Compare to the [Resolution](#) property.

Valid Values: 8, 16, 32

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Acquisition:Resolution Property

Short Name: Resolution

Property of [niScope](#)

Indicates the actual resolution in bits of valid data (as opposed to padding bits) in the acquired waveform. Compare to the [Binary Sample Width](#) property.

Valid Values: 8 to 32

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Acquisition:Acquisition Type Property

Short Name: Acquisition Type

Property of [niScope](#)

Specifies how the oscilloscope acquires data and fills the waveform record.

Normal (0)	Sets the oscilloscope to normal resolution mode. The oscilloscope can use real-time sampling or equivalent-time sampling.
Flex Res (1001)	Sets legacy oscilloscopes to flexible resolution mode, if supported.
DDC (1002)	Sets legacy oscilloscopes to DDC mode, if supported. To use DDC mode for the PXI/PCI-5142 or PXIe-5622, set this property to Normal and set the DDC Enabled property to True.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	niScope Configure Acquisition
Channel-based	No
Resettable	No

Acquisition:Sample Mode Property

Short Name: Sample Mode

Property of [niScope](#)

Returns the sample mode the digitizer is currently using.


Defined Values

Real Time (0)

Equivalent Time (1)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	niScope Actual Sample Mode
Channel-based	No
Resettable	No

Acquisition:Advanced:Enable RIS in Auto Setup Property


Short Name: Enable RIS in Auto Setup

Property of [niScope](#)

Indicates whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Fetch Record Number Property

Short Name: Fetch Record Number

Property of [niScope](#)

Sets the record to fetch. The record is from a channel you specify. The default value is 0.

Valid Values: Values greater than or equal to 0

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Fetch Number of Records Property

Short Name: Fetch Number of Records

Property of [niScope](#)

Fetches multiple records. If you want to fetch all records from the record you specify in the [Fetch Record Number](#) property to the last record configured, use -1. The default value is -1.

Related topics:

- [Making Multiple-Record Acquisitions](#)
- [Fetching Multiple-Record Acquisitions](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Fetch Relative To Property

Short Name: Fetch Relative To

Property of [niScope](#)

Specifies which point in the acquired waveform is the first to be fetched. This property specifies what the 'Fetch Offset' is relative to.

Start (482)	Fetch data starting at the first point sampled by the digitizer.
Trigger (483)	Fetch at the first posttrigger sample.
Pretrigger (477)	Fetches relative to the first pretrigger point requested with the niScope Configure Horizontal Timing VI.
Now (481)	Fetch data at the last sample acquired.
Read Pointer (388)	The read pointer is set to zero when a new acquisition is initiated. After every fetch the read pointer is incremented to be the sample after the last sample retrieved. Therefore, you can repeatedly fetch relative to the read pointer for a continuous acquisition program.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No

Fetch:Fetch Offset Property

Resetable

No

Short Name: Fetch Offset

Property of [niScope](#)

Sets the offset in samples; the samples returned also depend on the [Fetch Relative To](#) property. The default value is 0.

Valid Values: All integers

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	No

Fetch:Data Transfer Block Size Property

Short Name: Data Transfer Block Size

Property of [niScope](#)

Specifies the maximum number of samples to transfer at one time from the device to host memory. Increasing this number should result in better fetching performance because the driver does not need to restart the transfers as often. However, increasing this number may also increase the amount of page-locked memory required from the system.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Fetch Meas Num Samples Property


Short Name: Fetch Meas Number of Samples

Property of [niScope](#)

Determines the number of samples to fetch from a digitizer when performing a measurement. -1 means fetch all samples from the [Fetch Offset](#) property to the end of the current record. The default value is -1.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Points Done Property


Short Name: Points Done

Property of [niScope](#)

Actual number of samples acquired relative to the configured value for [Fetch Relative To](#), including [Fetch Offset](#), and for the current configured [Fetch Record Number](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only

High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Fetch Backlog Property

Short Name: Fetch Backlog

Property of [niScope](#)


Specifies the number of points acquired that have not been fetched yet.

Related topics:

- [Fetching Data](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Fetch:Records Done Property


Short Name: Records Done

Property of [niScope](#)

Returns the number of records your digitizer has acquired.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A

Channel-based	No
Resetable	No

Fetch:Advanced:Maximum Bandwidth Property

Short Name: Maximum Bandwidth

Property of [niScope](#)


Specifies the maximum bandwidth that the device is allowed to consume. The NI device limits itself to transfer fewer bytes per second on the PCIe bus than the value you specify for this property.

Related topics:

- [Bandwidth](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	No

Fetch:Advanced:Preferred Packet Size Property

Short Name: Preferred Packet Size

Property of [niScope](#)

Specifies the preferred size of the data field in the PCI Express packet. In general, the larger the packet size, the more efficiently the device uses the bus. However, some systems, because of their implementation, perform better with smaller packet sizes. The value of this property must be a power of two (64, 128, ... , 512).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking:Reference (Input) Clock Source Property

Short Name: Reference Clock Source

Property of [niScope](#)

Specifies the input source for the PLL reference clock.

Defined Values

VAL_NO_SOURCE

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PXI_CLOCK

VAL_CLK_IN

VAL_EXTERNAL

VAL_RTSM_CLOCK

Related topics:

- [Reference Clock/Phase-Lock Loop](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write

High-level VIs	niScope Configure Clock
Channel-based	No
Resetttable	No

Clocking:Reference Clock Rate Property

Short Name: Reference Clock Rate

Property of [niScope](#)

If [Reference Clock Source](#) is an external source, specifies the frequency (in hertz) of the input clock (reference clock) to which the internal sample clock timebase is synchronized.



Note Refer to [Features Supported by Device](#) for valid values.

Related topics:

- [Reference Clock/Phase-Lock Loop](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetttable	No

Clocking:Output Clock Source Property

Short Name: Output Clock Source

Property of [niScope](#)

Specifies the output source for the 10 MHz clock to which the sample clock of another digitizer can be phase-locked.

Defined Values

None

VAL_RTSI_CLOCK

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_CLK_OUT

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3


VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Clock
Channel-based	No
Resettable	Yes

Clocking: Sample Clock Timebase Source Property

Short Name: Sample Clock Timebase Source

Property of [niScope](#)

Specifies the source of the sample clock timebase, which is the timebase used to control waveform sampling.

The actual sample rate may be the timebase itself or a scaled version of the timebase, depending on the [Min Sample Rate](#) property (for internal sources) or the [Sample Clock Timebase Divisor](#) and [Sample Clock Timebase Multiplier](#) properties (for external sources).

Defined Values

VAL_CLK_IN

VAL_PXI_STAR

VAL_PFI_0

VAL_PFI_1

VAL_NO_SOURCE


VAL_ONBOARD_CONFIGURABLE_RATE_CLK

Related topics:

- [Sample Clock](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	Yes

Clocking: Sample Clock Timebase Rate Property

Short Name: Sample Clock Timebase Rate


Property of [niScope](#)

Specifies, in hertz, the frequency of the external clock used as the timebase source if the [Sample Clock Timebase Source](#) is an external source or, if the timebase source is

the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking: Sample Clock Timebase Divisor Property

Short Name: Sample Clock Timebase Divisor

Property of [niScope](#)


If [Sample Clock Timebase Source](#) is an external source, specifies the ratio between the sample clock timebase rate and the actual sample rate, which can be slower.

Related topics:

- [Sample Clock](#)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking:Sample Clock Timebase Multiplier Property

Short Name: Sample Clock Timebase Multiplier

Property of [niScope](#)

If [Sample Clock Timebase Source](#) is an external source, this property specifies the ratio between the [Sample Clock Timebase Rate](#) and the actual sample rate, which can be higher. This property can be used in conjunction with the [Sample Clock Timebase Divisor Property](#).

Some devices use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this property vary by device and current configuration.

Related topics:

- [Sample Clock](#)

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking:Exported Sample Clock Output Terminal Property

Short Name: Exported Sample Clock Output Terminal

Property of [niScope](#)

Exports the sample clock to a specified terminal. This property is not supported by all digitizers.

The full sample clock rate can be exported to the CLK_OUT connector. If decimating, the divided down sample clock rate can be exported to any of the valid destinations.

Defined Values

VAL_CLK_OUT

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PXI_STAR


VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Clocking:Clock Sync Pulse Source Property

Short Name: Clock Sync Pulse Source

Property of [niScope](#)

Specifies the line on which the sample clock or the one-time sync pulse is sent or received.

This line should be the same for all devices to be synchronized.

Defined Values

VAL_NO_SOURCE

VAL_RTISI_0

VAL_RTISI_1

VAL_RTISI_2

VAL_RTISI_3

VAL_RTISI_4

VAL_RTISI_5


VAL_RTISI_6

VAL_PFI_1

VAL_PFI_2

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Clock
Channel-based	No
Resettable	No

Clocking:PLL Lock Status Property

Short Name: PLL Locked

Property of [niScope](#)

If TRUE, the PLL has remained locked to the external reference clock since it was last checked. If FALSE, the PLL has become unlocked from the external reference clock since it was last checked.

Remarks

The following table lists the characteristics of this property.

Datatype	<input checked="" type="checkbox"/> TRUE
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking:Advanced:Oscillator Phase DAC Value Property

Short Name: Oscillator Phase DAC Value

Property of [niScope](#)

Gets or sets the binary phase DAC value that controls the delay added to the phase-locked loop (PLL) of the sample clock.

Valid Values: [0, 65535]

Default Value: " "

Use this property when performing manual adjustment to correct for skew and jitter between oscilloscopes synchronized with NI-TClk.

1. Apply time offset with the [Adjust Sample Clock Relative Delay VI](#).

2. Use this property to convert the time offset into a value that can apply the manual adjustment across sessions and improve synchronization repeatability.

For details on performing manual adjustment, refer to [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#).



Note If this value is set, sample clock adjustments and TClk cannot do any subsample adjustment of the timebase sample clock.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Clocking:Advanced:Absolute Sample Clock Offset Property

Short Name: Absolute Sample Clock Offset

Property of [niScope](#)

Gets or sets the absolute time offset, in seconds, of the sample clock relative to the reference clock.

Valid Values: [-0.5 **Sample Clock Period**, 0.5 **Sample Clock Period**]

Default Value: 0


Use this property when performing manual adjustment to correct for skew and jitter between oscilloscopes synchronized with NI-TClk.

Apply offset with this property to reduce skew and jitter, and write the value you set to apply the manual adjustment across sessions and improve synchronization repeatability.

For details on performing manual adjustment, refer to [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Start Trigger (Acq. Arm): Source Property

Short Name: Start Trigger Source

Property of [niScope](#)

Specifies the source the digitizer monitors for an acquisition arm trigger. When an acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Defined Values

VAL_IMMEDIATE

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1
 VAL_PFI_2
 VAL_PFI_3
 VAL_PFI_4
 VAL_PFI_5
 VAL_PFI_6
 VAL_PFI_7
 VAL_PXI_STAR
 VAL_SW_TRIG_FUNC

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Start Trigger (Acq. Arm): Output Terminal Property

Short Name: StartTrig.OutputTerm

Property of [niScope](#)

Specifies the destination to export the Start trigger. When the start trigger is received, the digitizer begins acquiring data. Refer to the device specifications document for a list of valid destinations.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:Start Trigger (Acq. Arm):Terminal Name Property

Short Name: Start Trigger Terminal Name

Property of [niScope](#)

Returns the fully qualified name for the Start Trigger terminal.

You can use this terminal as the source for another trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Ready for Start:Output Terminal Property

Short Name: RdyForStartEvent.OutputTerm

Property of [niScope](#)

Specifies the destination to export the Start trigger. When the start trigger is received, the digitizer begins acquiring data. Refer to the device specifications document for a list of valid destinations.

Defined Values

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5


VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:Ready for Start:Terminal Name Property

Short Name: Ready for Start Event Terminal Name

Property of [niScope](#)

Returns the fully qualified name for the Ready for Start Event terminal.

You can use this terminal as the source for a trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Advance Trigger:Source Property

Short Name: Advance Trigger Source

Property of [niScope](#)

Specifies the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples for the next record.

Defined Values

VAL_IMMEDIATE

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5

VAL_PFI_6


VAL_PFI_7

VAL_PXI_STAR

VAL_SW_TRIG_FUNC

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Advance Trigger:Output Terminal Property


Short Name: AdvTrig.OutputTerm

Property of [niScope](#)

Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Advance Trigger:Terminal Name Property

Short Name: Advance Trigger Terminal Name


Property of [niScope](#)

Returns the fully qualified name for the Advance Trigger terminal.

You can use this terminal as the source for another trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A

Channel-based	No
Resettable	No

Synchronization:Ready for Advance:Output Terminal Property

Short Name: RdyForAdvEvent.OutputTerm

Property of [niScope](#)

Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Defined Values

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5


VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:Ready for Advance:Terminal Name Property

Short Name: Ready for Advance Event Terminal Name


Property of [niScope](#)

Returns the fully qualified name for the Ready for Advance Event terminal.

You can use this terminal as the source for a trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Arm Reference Trigger:Source Property

Short Name: Arm Reference Trigger Source

Property of [niScope](#)

Specifies the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins searching for the reference (stop) trigger from the user-configured trigger source.

Defined Values

VAL_IMMEDIATE

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5

VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

VAL_SW_TRIG_FUNC

Remarks

The following table lists the characteristics of this property.

Datatype	abc
----------	-----

Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:Ready for Reference:Output Terminal Property


Short Name: RdyForRefEvent.OutputTerm

Property of [niScope](#)

Specifies the destination for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger. Refer to the device-specific documentation in the **NI High-Speed Digitizers Help** for a list of valid destinations for your device.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:Ready for Reference:Terminal Name Property

Short Name: Ready for Reference Event Terminal Name

Property of [niScope](#)

Returns the fully qualified name for the Ready for Reference Event terminal.

You can use this terminal as the source for a trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:End of Record:Output Terminal Property

Short Name: EndOfRecEvent.OutputTerm

Property of [niScope](#)

Specifies the destination for the End of Record event. When this event is asserted, the digitizer has completed sampling a record. Refer to the device specifications document for a list of valid destinations.

Defined Values

VAL_RTISI_0

VAL_RTISI_1

VAL_RTISI_2

VAL_RTISI_3

VAL_RTISI_4

VAL_RTISI_5

VAL_RTISI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5

VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:End of Record:Terminal Name Property

Short Name: End of Record Event Terminal Name

Property of [niScope](#)

Returns the fully qualified name for the End of Record Event terminal.

You can use this terminal as the source for a trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No

Synchronization:End of Acquisition:Output Terminal Property

Short Name: EndOfAcqEvent.OutputTerm

Property of [niScope](#)

Specifies the destination for the End of Acquisition event. When this event is asserted, the digitizer has completed sampling all records. Refer to the device specifications document for a list of valid destinations.

Defined Values

VAL_RTSI_0

VAL_RTSI_1

VAL_RTSI_2

VAL_RTSI_3

VAL_RTSI_4

VAL_RTSI_5

VAL_RTSI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PFI_3

VAL_PFI_4

VAL_PFI_5


VAL_PFI_6

VAL_PFI_7

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization:End of Acquisition:Terminal Name Property

Short Name: End of Acquisition Event Terminal Name


Property of [niScope](#)

Returns the fully qualified name for the End of Acquisition Event terminal.

You can use this terminal as the source for a trigger.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Synchronization:5 Volt Power:Output Terminal Property

Short Name: 5VoltPower.OutputTerm

Property of [niScope](#)

Specifies the destination for the 5 Volt power signal. Refer to the device specifications document for a list of valid destinations.



Note This property is supported only for NI 5152/5153/5154 devices.

Defined Values

VAL_RTISI_0

VAL_RTISI_1

VAL_RTISI_2

VAL_RTISI_3

VAL_RTISI_4

VAL_RTISI_5

VAL_RTISI_6

VAL_PFI_0

VAL_PFI_1

VAL_PFI_2

VAL_PXI_STAR

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Export Signal
Channel-based	No
Resettable	No

Synchronization: Master Enable Property

Short Name: Master Enable

Property of [niScope](#)

Specifies whether the device is a master or a slave.

The master device is typically the originator of the trigger signal and clock sync pulse. For a stand-alone device, set this property to FALSE.


Valid Range

TRUE—Master

FALSE—Slave

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Configure Clock
Channel-based	No
Resettable	No

Waveform Measurement: Other Channel Property


Short Name: Other Channel

Property of [niScope](#)

Specifies the second channel for two-channel measurements, such as [Add Channels](#). If processing steps are registered with this channel, the processing happens before the waveform is used in a two-channel measurement. The default value is 0.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Resettable	No

Waveform Measurement:Array Gain Property


Short Name: Array Gain

Property of [niScope](#)

Every element of an array is multiplied by this scalar value during the array gain measurement. The default value is 1.0.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	Yes

Waveform Measurement:Array Offset Property


Short Name: Array Offset

Property of [niScope](#)

Every element of an array is added to this scalar value during the array offset measurement. The default value is 0.0.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Hysteresis Percent Property


Short Name: Hysteresis Percent

Property of [niScope](#)

Digital hysteresis that is used in several of the scalar waveform measurements. This property specifies the percentage of the full-scale vertical range for the hysteresis window size. The default value is 2%.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement>Last Acq. Histogram Size Property


Short Name: Last Acq. Histogram Size

Property of [niScope](#)

Specifies the size (that is, the number of bins) in the last acquisition histogram. This histogram is used to determine several scalar measurements, most importantly voltage low and voltage high. The default value is 256.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write

High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Voltage Histogram:Size Property

Short Name: V. Hist. Size

Property of [niScope](#)

Specifies the number of bins in the running voltage histogram. The default value is 256.

This value is used during the first running voltage histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Voltage Histogram:Low Volts Property

Short Name: V. Hist. Low Volts


Property of [niScope](#)

Specifies the minimum voltage value in the running voltage histogram. The default value is -10.0.

This value is used during the first running voltage histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Voltage Histogram:High Volts Property

Short Name: V. Hist. High Volts


Property of [niScope](#)

Specifies the maximum voltage value in the running voltage histogram. The default value is 10.0.

This value is used during the first running voltage histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Time Histogram:Size Property


Short Name: Time Hist. Size

Property of [niScope](#)

Determines the multiple acquisition time histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement: Time Histogram: Low Volts Property

Short Name: Time Hist. Low Volts


Property of [niScope](#)

Specifies the low voltage limit for the multi-acquisition time histogram. Only points in the waveform between the low and high voltage limits are included in the histogram. The default value is -10.0.

This value is used during the first running time histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes

Waveform Measurement: Time Histogram: High Volts Property

Short Name: Time Hist. High Volts


Property of [niScope](#)

Specifies the high voltage limit for the Multi-Acquisition time histogram. Only points in the waveform between the low and high voltage limits are included in the histogram. The default value is 10.0 V.

This value is used during the first time histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement: Time Histogram: Low Time Property

Short Name: Time Hist. Low Time


Property of [niScope](#)

Specifies the minimum time limit (in seconds) of the multi-acquisition time histogram, where the time is in seconds relative to the trigger position. Only points in the waveform between the low and high time limits are included in the histogram. The default value is -5.0e-4 .

This value is used during the first time histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement: Time Histogram: High Time Property

Short Name: Time Hist. High Time


Property of [niScope](#)

Specifies the maximum time limit (in seconds) of the Multi-Acquisition time histogram, where the time is in seconds relative to the trigger position. Only points in the waveform between the low and high time limits are included in the histogram. The default value is 5.0e-4 .

This value is used during the first time histogram measurement, and it is not updated until you call [niScope Clear Waveform Measurement Stats](#).

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Interpolation:Polynomial Interpolation Order Property

Short Name: Poly. Interp. Order

Property of [niScope](#)

Specifies the order of the polynomial used during the polynomial interpolation array measurement. For example, an order of 1 is linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid. The default value is 1 (linear interpolation).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Interpolation:Sampling Factor Property

Short Name: Interp. Sampling Factor

Property of [niScope](#)

The new number of points for polynomial interpolation is the sampling factor times the input number of points. The default value is 0.0.

For example, if you acquire 1,000 points with the digitizer and set this property to 2.5, calling the [niScope Fetch Measurement \(poly\)](#) VI (Measurement Scalar DBL

instance), with the Polynomial Interpolation measurement resamples the waveform to 2,500 points.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Filter:Type Property

Short Name: Filter Type

Property of [niScope](#)

Specifies the type of digital filter. The default value is lowpass.

lowpass (0)	Specifies lowpass as the filter type.
highpass (1)	Specifies highpass as the filter type.
bandpass (2)	Specifies bandpass as the filter type.
bandstop (3)	Specifies bandstop as the filter type.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Filter:FIR Taps Property

Short Name: Filter Taps

Property of [niScope](#)

Specifies the number of taps for the finite impulse response filter. This value must be odd if the filter type is highpass or bandstop. Otherwise, the magnitude response goes to zero as the frequency goes to half the sampling rate. The default value is 25.

Valid Values: >0

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Filter:FIR Window Property

Short Name: FIR Filter Window

Property of [niScope](#)


Specifies the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters. The default value is None (0).

None (0)	No window.
Hanning (409)	Specifies a Hanning window.
Hamming (420)	Specifies a Hamming window.
Flat Top (410)	Specifies a Flat Top window.

Triangle (423)	Specifies a Triangle window.
Blackman (424)	Specifies a Blackman window.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Filter:Width Property


Short Name: Filter Width

Property of [niScope](#)

Specifies the width (in Hz) of a bandpass or bandstop filter. The cutoff frequencies are the (center frequency property $\pm 0.5 \times$ filter width). The default value is 1.0e3.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Filter:IIR Order Property

Short Name: Filter Order

Property of [niScope](#)

Specifies the order of the infinite impulse response filter. The default value is 2.

Valid Values: >0

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Filter:Cutoff Frequency Property

Short Name: Filter Cutoff Frequency

Property of [niScope](#)

Specifies the cutoff frequency in hertz for filters of type lowpass and highpass. The cutoff frequency definition varies depending on the filter. The default value is 1.0e6 Hz.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Filter:Center Frequency Property


Short Name: Filter Center Frequency

Property of [niScope](#)

The center frequency in hertz for filters of type bandpass and bandstop. The width of the filter is specified by Filter Width, where the cutoff frequencies are the center width. The default value is 1.0e6 Hz.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Filter:Ripple Property

Short Name: Filter Ripple


Property of [niScope](#)

Specifies the amount of passband ripple (in dB) for Chebyshev filters. More ripple gives a sharper cutoff for a given filter order. The default value is 0.1.

Valid Values: >0.0

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Waveform Measurement:Filter:Percent Waveform Transient Property

Short Name: Percent Waveform Transient


Property of [niScope](#)

The percentage (0 - 100%) of the infinite impulse response (IIR) filtered waveform to eliminate from the beginning of the waveform. This action allows eliminating the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition. The default value is 20.0%.

Valid Range: 0.0 - 100.0%

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Reference Levels:Channel Based Low Ref Level Property

Short Name: Chan. Based Low Ref


Property of [niScope](#)

Specifies the low reference level used in many scalar measurements. The default value is 10.0%.

Units: Percentage of the signal based on the selected [Percentage Units Method](#) property.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Reference
Levels:Channel Based Mid Ref Level Property

Short Name: Chan. Based Mid Ref


Property of [niScope](#)

Specifies the mid reference level used in many scalar measurements. The default value is 50%.

Units: Percentage of the signal based on the selected [Percentage Units Method](#) property.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Reference
Levels:Channel Based High Ref Level Property

Short Name: Chan Based High Ref


Property of [niScope](#)

Specifies the high reference level used in many scalar measurements. The default value is 90%.

Units: Percentage of the signal based on the selected [Percentage Units Method](#) property.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Reference Levels:Units Property

Short Name: Ref. Level Units

Property of [niScope](#)

Specifies the units for the waveform measurement reference levels.

If you choose percentage, the measurement routine uses the [Percentage Method](#) property to map the percentage values to voltages. If you choose voltage units, you can set the voltage thresholds directly and avoid extra calculations.

Volts (0)	Specifies that the reference levels are given in units of volts.
Percentage (1)	(Default) Specifies that the reference levels are given in percentage units.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Waveform Measurement:Reference Levels:Percentage Units Method Property

Short Name: Percentage Method

Property of [niScope](#)

Specifies the method used to map percentage reference units to voltages. The default value is BaseTop.

LowHigh (0)	Specifies that the reference level percentages should be computed using the low/high method,
MinMax (1)	Reference level percentages are computed using the min/max method.
BaseTop (2)	Reference level percentages are computed using the base/top method.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:DDC:DDC Enabled Property

Short Name: DDC Enabled

Property of [niScope](#)

Enables/disables the digital downconverter (DDC) block of the digitizer. When the DDC block is disabled, all DDC-related properties are disabled and have no effect on the acquired signal. The default value is FALSE.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:DDC:Data Processing Mode Property

Short Name: Data Processing Mode

Property of [niScope](#)

The way in which data is processed by the DDC block. The default value is Complex.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Real (0)	The waveform data points are real numbers (I data).
Complex (1)	The waveform data points are complex numbers (IQ data).

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Onboard Signal Processing:DDC:Frequency Translation Enabled Property

Short Name: Frequency Translation Enabled

Property of [niScope](#)

Enables/disables frequency translating the data around the user-selected center frequency down to baseband. The default value is TRUE.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:DDC:Center Frequency Property

Short Name: Center Frequency

Property of [niScope](#)

The frequency at which the [DDC](#) block frequency translates the input data. The default value is 10 MHz.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Valid Values

0 - (0.5 × Sample Clock Timebase Rate for digitizer)

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No


Onboard Signal Processing:DDC:Fetch Interleaved IQ Data Property

Short Name: Fetch Interleaved IQ Data

Property of [niScope](#)


Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms. If enabled, the number of elements returned by scalar fetch types (such as 16-bit integer) is twice the requested number of samples. If

disabled during DDC acquisitions in Complex mode, two noninterleaved arrays of data are returned per channel, per record.

 **Note** This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.


Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No


Onboard Signal Processing:DDC:Q Source Property

Short Name: Q Source

Property of [niScope](#)

Specifies the channel that is the input to the Q data stream of the [DDC](#).

 **Note** For multi-instrument sessions, the channel you define with this property as the Q data stream and the channel for the corresponding I data stream must both be located within the same physical instrument.


 **Note** This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Valid Values: All valid channels for the device

Default Value: The channel to which the property is registered

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Frequency Translation

Frequency Translation Phase I

Onboard Signal Processing:DDC:Signal Adjustments:Frequency Translation:Frequency Translation Phase I Property

Short Name: Frequency Translation Phase I

Property of [niScope](#)

The I oscillator phase in degrees at the first point acquired. The default value is 0.

Valid Values


-360 to 360



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A

Channel-based	Yes
Resettable	No

Frequency Translation Phase Q

Onboard Signal Processing:DDC:Signal Adjustments:Frequency Translation:Frequency Translation Phase Q Property

Short Name: Frequency Translation Phase Q

Property of [niScope](#)

The Q oscillator phase in degrees at the first point acquired. Use this property only when the [Data Processing Mode](#) property is set to Complex. The default value is 90.

Valid Values

-360 to 360



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	DBL
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:Equalization:Equalization Filter Enabled Property

Short Name: Equalization Filter Enabled

Property of [niScope](#)

Enables the onboard signal processing equalization FIR block, which is connected directly to the input signal. The equalization filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1. The default value is FALSE.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:Equalization:Equalization Num Coefficients Property

Short Name: Equalization Num Coefficients

Property of [niScope](#)

Returns the number of coefficients that the equalization FIR filter can accept. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	Yes
Resettable	No

Onboard Signal Processing:Fractional Resample:Fractional Resample Enabled Property

Short Name: Fractional Resample Enabled

Property of [niScope](#)

Enables the onboard signal processing block that resamples the input waveform to the user desired sample rate. The default value is FALSE.



Note This property can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this property with a device that does not support OSP.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Onboard Signal Processing:OSP Overflow Error Reporting Property

Short Name: Overflow Error Reporting

Property of [niScope](#)

Configures error reporting when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform. The default value is Warning.

Error (0)	Execution stops and NI-SCOPE returns an error when an overflow has occurred in the OSP block.
Warning (1)	Execution continues and NI-SCOPE returns a warning when an overflow has occurred in the OSP block.
Disabled (2)	NI-SCOPE does not return an error when an overflow has occurred in the OSP block.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:P2P Enabled Property

Short Name: P2P Enabled

Property of [niScope](#)

Specifies whether the digitizer writes data to the peer-to-peer endpoint. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Channels to Stream Property

Short Name: Channels to Stream

Property of [niScope](#)

Specifies which channels will be written to a peer-to-peer endpoint. If multiple channels are specified, they will be interleaved by sample. This property is endpoint-based. The default value is 0.




Note This property can only be used with high-speed digitizers that support peer-to-peer streaming.



Note This property must either be unused or set to all enabled channels on NI 5160/5162 digitizers.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Endpoint Size Property

Short Name: Endpoint Size

Property of [niScope](#)


Returns the size, in samples, of the peer-to-peer endpoint. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.


Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Samples Available In Endpoint Property

Short Name: Samples Available In Endpoint

Property of [niScope](#)

Returns the current number of samples available to stream from a peer-to-peer endpoint. This property is endpoint-based.

 **Note** This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.


Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Most Samples Available in Endpoint Property

Short Name: Most Samples Available in Endpoint

Property of [niScope](#)

Returns the most number of samples available to stream from a peer-to-peer endpoint since the last time this property was read. This property is endpoint-based.

 **Note** This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No

Peer-to-Peer:Samples Transferred Property

Resetable

No

Short Name: Samples Transferred

Property of [niScope](#)

Returns the number of samples transferred through the peer-to-peer endpoint since the endpoint was last reset. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Peer-to-Peer:Endpoint Overflow Property

Short Name: Endpoint Overflow

Property of [niScope](#)

Returns TRUE if the peer-to-peer endpoint has overflowed. Reset the endpoint to clear the overflow condition. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	TF
----------	-----------

Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:FIFO Endpoint Count Property

Short Name: FIFO Endpoint Count

Property of [niScope](#)

Returns the number of FIFO-based peer-to-peer endpoints this device supports.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Onboard Memory Enabled Property

Short Name: Onboard Memory Enabled

Property of [niScope](#)

Specifies whether the digitizer writes data to onboard memory when a peer-to-peer endpoint is enabled.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.



Note This property is not supported on NI 5160/5162 digitizers.

Remarks

The following table lists the characteristics of this property.


Datatype	TF1
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Stream Relative To Property

Short Name: Stream Relative To

Property of [niScope](#)

Determines which trigger peer-to-peer data is streamed relative to. The default value is Start Trigger.

 Note On the NI 5122/5622, only Start Trigger is valid for this property.	
Start Trigger (0)	Data is streamed from the start trigger.
Reference Trigger (1)	Data is streamed relative to the reference trigger and reference position.
Sync Trigger (2)	Data is streamed relative to the sync trigger and reference position.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Samples Transferred Per Record Property

Short Name: Samples Transferred Per Record

Property of [niScope](#)

Returns the number of samples transferred per record when you set the [Stream Relative To](#) property to Reference Trigger or Sync Trigger.



Note This property is only supported on NI 5160/5162 digitizers.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Peer-to-Peer:Manual:Manual Configuration Enabled Property

Short Name: Manual Configuration Enabled

Property of [niScope](#)


Enables and disables manual configuration of a peer-to-peer endpoint. These attributes cannot be used if an endpoint is being configured by NI-P2P, or a resource reservation error will result. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Data Transfer Permission Address

Peer-to-Peer:Manual:Configuration:Data Transfer Permission Address Property

Short Name: Data Transfer Permission Address

Property of [niScope](#)


Returns the address of a hardware register used to grant permission for the peer-to-peer endpoint to write data to another peer. The type of this address is determined by the [Data Transfer Permission Address Type](#) property. Permission is granted in bytes and the register is additive. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Data Transfer Permission Address Type

Peer-to-Peer:Manual:Configuration:Data Transfer Permission Address Type Property

Short Name: Data Transfer Permission Address Type

Property of [niScope](#)

Specifies the type of address returned to the user from the [Data Transfer Permission Address](#) property. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Virtual (1)	Virtual address.
Physical (0)	Physical address.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No


Destination Window Address

Peer-to-Peer:Manual:Configuration:Destination Window Address Property

Short Name: Destination Window Address

Property of [niScope](#)

Specifies the destination for data written by the peer-to-peer endpoint. The type of this address is specified by the [Destination Window Address Type](#) property. This property is endpoint-based.

 Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Valid Values

A valid, non-NULL, physical or virtual address.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	No


Destination Window Address Type

Peer-to-Peer:Manual:Configuration:Destination Window Address Type Property

Short Name: Destination Window Address Type

Property of [niScope](#)

Specifies the type of the [Destination Window Address](#) property. This property is endpoint-based.

 Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Virtual (1)	Virtual address.
Physical (0)	Physical address.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	No

Destination Window Size

Peer-to-Peer:Manual:Configuration:Destination Window Size Property

Short Name: Destination Window Size

Property of [niScope](#)

Specifies the size, in bytes, of the destination window determined by the [Destination Window Address](#) and [Destination Window Address Type](#) properties. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resetable	No

Push Message On

Peer-to-Peer:Manual:Notification:Push Message On Property

Short Name: Push Message On

Property of [niScope](#)

Specifies the event to push the [Message Push Value](#) property to the [Message Push Address](#) property. Specifying Done will push the message when the acquisition has completed. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Done (1)	Notify when digitizer acquisition is done.
Never (0)	Never send notification.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No


Message Push Address

Peer-to-Peer:Manual:Notification:Message Push Address Property

Short Name: Message Push Address

Property of [niScope](#)

Specifies the address to push the [Message Push Value](#) to on the event specified by the [Push Message On](#) property. This property is endpoint-based.

 Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No


Message Push Address Type

Peer-to-Peer:Manual:Notification:Message Push Address Type Property

Short Name: Message Push Address Type

Property of [niScope](#)

Specifies the type of the [Message Push Address](#) property. This property is endpoint-based.

 Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Virtual (1)	Virtual address.
Physical (0)	Physical address.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
----------	------------

Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Message Push Value

Peer-to-Peer:Manual:Notification:Message Push Value Property

Short Name: Message Push Value

Property of [niScope](#)

Specifies the value to be pushed to the [Message Push Address](#) property on the event specified in the [Push Message On](#) property. This property is endpoint-based.



Note This property can be used only with high-speed digitizers that support peer-to-peer streaming.

Remarks

The following table lists the characteristics of this property.

Datatype	I64
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:User Options:Range Check Property

Short Name: Range Check


Property of [niScope](#)

Specifies whether to validate property values and function parameters. If enabled, the instrument driver validates the parameter values that you pass to driver

functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this property to FALSE to disable range checking and maximize performance. The default value is TRUE. Use [niScope Initialize with Options](#) to override this value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Initialize With Options
Channel-based	No
Resetable	No

Inherent IVI Attributes: User Options: Query Instrument Status Property


Short Name: Query Instrument Status

Property of [niScope](#)

Specifies whether the instrument driver queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this property to FALSE to disable status checking and maximize performance. The instrument driver can choose to ignore status checking for particular properties regardless of the setting of this property. The default value is TRUE. Use [niScope Initialize with Options](#) to override this value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Initialize With Options
Channel-based	No

Inherent IVI Attributes:User Options:Cache Property


Short Name: Cache

Property of [niScope](#)

Specifies whether to cache the value of properties. When caching is enabled, the instrument driver keeps track of the current instrument settings and avoids sending redundant commands to the instrument. Thus, you can significantly increase execution speed. The instrument driver can choose always to cache or never to cache particular properties, regardless of the setting of this property. The default value is TRUE. Use [niScope Initialize with Options](#) to override this value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Initialize With Options
Channel-based	No
Resetable	Yes

Inherent IVI Attributes:User Options:Simulate Property


Short Name: Simulate

Property of [niScope](#)

Specifies whether to simulate instrument driver I/O operations. The default value is FALSE. Use [niScope Initialize with Options](#) to override this value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Initialize With Options
Channel-based	No
Resettable	Yes

Inherent IVI Attributes:User Options:Record Value Coercions Property


Short Name: Record Value Coercions

Property of [niScope](#)

Specifies whether the IVI engine keeps a list of the value coercions it makes for ViInt32 and DBL properties. The default value is FALSE. Use [niScope Initialize with Options](#) to override this value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	niScope Initialize With Options
Channel-based	No
Resettable	No

Inherent IVI Attributes:User Options:Interchange Check Property


Short Name: Interchange Check

Property of [niScope](#)

Specifies whether to perform interchangeability checking and log interchangeability warnings when you call VIs. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:Driver
Identification:Description Property


Short Name: Description

Property of [niScope](#)

A string that contains the description of the instrument.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:Driver Identification:Driver
Prefix Property


Short Name: Driver Prefix

Property of [niScope](#)

A string that contains the prefix for the instrument driver. The name of each user-callable function in this driver starts with this prefix.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes: Driver Identification: Driver Vendor Property


Short Name: Driver Vendor

Property of [niScope](#)

A string that contains the name of the vendor that supplies this driver, for example, "National Instruments".

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:Driver Identification:Revision Property

Short Name: Revision

Property of [niScope](#)

The string that contains additional version information about this instrument driver.

Remarks

The following table lists the characteristics of this property.

Datatype	<code>abc</code>
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:Driver Identification:Class Specification Major Version Property

Short Name: Class Specification Major Version

Property of [niScope](#)

The major version number of the class specification with which this driver is compliant.

Remarks

The following table lists the characteristics of this property.

Datatype	<code>I32</code>
Permissions	Read Only
High-level VIs	N/A
Channel-based	No

Inherent IVI Attributes:Driver Identification:Class Specification Minor Version Property

Short Name: Class Specification Minor Version

Property of [niScope](#)

The minor version number of the class specification with which this driver is compliant.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Inherent IVI Attributes:Driver Capabilities:Supported Instrument Models Property

Short Name: Supported Instrument Models

Property of [niScope](#)

A string that contains a comma-separated list of the instrument model numbers supported by this driver.

Remarks

The following table lists the characteristics of this property.

Datatype	abc
----------	------------

Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Inherent IVI Attributes:Driver Capabilities:Class Group Capabilities Property


Short Name: Class Group Capabilities

Property of [niScope](#)

A string that contains a comma-separated list of class-extension groups that this driver implements.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Inherent IVI Attributes:Driver Capabilities:Channel Count Property

Short Name: Channel Count

Property of [niScope](#)

Indicates the number of channels that the specific instrument driver supports. For channel based properties, the IVI engine maintains a separate cache value for each channel.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Inherent IVI Attributes:Instrument Identification:Manufacturer Property

Short Name: Manufacturer

Property of [niScope](#)

A string that contains the name of the instrument manufacturer, for example, "National Instruments".

Remarks

The following table lists the characteristics of this property.

Datatype	abc
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resetable	No

Inherent IVI Attributes:Instrument Identification:Model Property


Short Name: Model

Property of [niScope](#)

A string that contains the model number of the current instrument.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes: Instrument Identification: Firmware Revision Property


Short Name: Firmware Revision

Property of [niScope](#)

A string that contains the firmware revision information for the current instrument.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes: Advanced Session Information: Logical Name Property

Short Name: Logical Name


Property of [niScope](#)

A string that contains the logical name you specified when opening the current IVI session.

You can pass a logical name to [niScope Initialize](#) or [niScope Initialize with Options](#). The IVI Configuration utility must contain an entry for the logical name. The logical name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Inherent IVI Attributes:Advanced Session Information:Resource Descriptor Property


Short Name: Resource Descriptor

Property of [niScope](#)

Indicates the resource descriptor the driver uses to identify the physical device. If you initialize the driver with a logical name, this property contains the resource descriptor that corresponds to the entry in the IVI Configuration utility. If you initialize the instrument driver with the resource descriptor, this property contains that value.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

Device Specific:IF Digitizer:Fetch Interleaved Data Property

Short Name: Fetch Interleaved Data

Property of [niScope](#)


Specifies whether to interleave I and Q data retrieved from the IF digitizer in a single array or to retrieve two separate arrays, one for I data and another for Q data.

Default Value: FALSE

TRUE	Retrieves a single array of alternating I and Q values. The resulting array is twice the size of the actual record length.
FALSE	Retrieves two arrays, one for I values and another for Q values.

Remarks

The following table lists the characteristics of this property.

Datatype	
Permissions	Read/Write
High-level VIs	N/A
Channel-based	No
Resettable	No

Device Specific:IF Digitizer:Device Number Property

Short Name: Device Number

Property of [niScope](#)

Returns the device number of the IF digitizer associated with the current session.

Remarks

The following table lists the characteristics of this property.

Datatype	I32
Permissions	Read Only
High-level VIs	N/A
Channel-based	No
Resettable	No

NI-SCOPE Function Reference Help

May 2020

This help file provides C/C++/VB programming support for NI-SCOPE, the National Instruments driver that communicates with your high-speed digitizer. This help file describes the NI-SCOPE functions and attributes you can use to configure and operate NI high-speed digitizers.

© 2001–2020 National Instruments Corporation. All rights reserved.

NI-SCOPE Function Tree

Expand this book to view the NI-SCOPE functions.

niScope_init

C Function Prototype

```
ViStatus niScope_init (ViRsrc resourceName, ViBoolean IDQuery, ViBoolean
resetDevice, ViSession* vi);
```

Purpose

Performs the following initialization actions:

- Creates a new IVI instrument driver session

- Opens a session to the specific driver using the interface and address (devices) you specify in the `resourceName`
- Queries each instrument ID and checks that it is valid for NI-SCOPE
- Resets the digitizer(s) to a known state, if `resetDevice` is set to `VI_TRUE`
- Sends initialization commands to set the instrument(s) to the state necessary for the operation of the instrument driver
- Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver function calls

Related topics:

- [NI-SCOPE Programming Flow](#)
- [NI-SCOPE Tutorial](#)

Parameters

Input

Name	Type	Description
<code>resourceName</code>	<code>ViRsrc</code>	Specifies the device name assigned by Measurement & Automation Explorer (MAX) to an NI-SCOPE instrument, for example, PXI1Slot3 , where <code>PXI1Slot3</code> is an instrument resource name.

This parameter accepts a comma-delimited list of strings in the form **PXI1Slot3,PXI1Slot4**, where `PXI1Slot3` is one instrument resource name and `PXI1Slot4` is another.



Note You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

resourceName Examples

Example#	Device Type	Syntax	Variable
1	NI-DAQmx device	myDAQmxDevice	(myDAQmxDevice = device name)
2	NI-DAQmx device	DAQ::myDAQmxDevice	(myDAQmxDevice = device name)

3	NI-DAQmx device	DAQ:: 2	(2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName	(myLogicalName = name)

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternate syntax for NI-DAQmx devices consists of DAQ::**NI-DAQmx device name**, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.

You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.



Note NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

idQuery

ViBoolean Specify whether to perform an ID query.

When you set this parameter to `VI_TRUE`, NI-SCOPE verifies that the device you initialize is a type that it supports.

When you set this parameter to `VI_FALSE`, the function initializes the device without performing an ID query.

Defined Values

`VI_TRUE`—Perform ID query

`VI_FALSE`—Skip ID query

Default Value: VI_TRUE

resetDevice ViBoolean Specify whether to reset the device during the initialization process.

Defined Values

VI_TRUE—Reset device

VI_FALSE—Do not reset device

Default Value: VI_TRUE

Output

Name	Type	Description
vi	ViSession*	Returns a session handle that you can use to identify the device in all subsequent NI-SCOPE function calls.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_InitWithOptions

C Function Prototype

```
ViStatus niScope_InitWithOptions (ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, ViString optionString, ViSession* newVi);
```

Purpose

Performs the following initialization actions:

- Creates a new IVI instrument driver session and optionally sets the initial state of the following session properties: Range Check, Cache, Simulate, Record Value Coercions
- Opens a session to the device(s) specified using the interface and address you use for the resourceName
- Queries each instrument ID and verifies that it is valid for this instrument driver
- Resets the digitizer(s) to a known state if resetDevice is set to VI_TRUE
- Returns an instrument handle that you use to identify the instrument(s) in all subsequent instrument driver function calls

Related topics:

- [NI-SCOPE Programming Flow](#)

Parameters

Input

Name	Type	Description
resourceName	ViRsrc	Specifies the device name assigned by Measurement & Automation Explorer (MAX) to an NI-SCOPE instrument, for example, PXI1Slot3 , where PXI1Slot3 is an instrument resource name.

This parameter accepts a comma-delimited list of strings in the form **PXI1Slot3,PXI1Slot4**, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.



Note You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

resourceName Examples

Example#	Device Type	Syntax	Variable
1	NI-DAQmx device	myDAQmxDevice	(myDAQmxDevice = device name)

2	NI-DAQmx device	DAQ:: myDAQmxDevice	(myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ:: 2	(2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName	(myLogicalName = name)

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternate syntax for NI-DAQmx devices consists of DAQ::**NI-DAQmx device name**, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.

You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.



Note NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration Store file exactly, without any variations in the case of the characters.

idQuery

ViBoolean Specify whether to perform an ID query.

When you set this parameter to `VI_TRUE`, NI-SCOPE verifies that the device you initialize is a type that it supports.

When you set this parameter to `VI_FALSE`, the function initializes the device without performing an ID query.

Defined Values

VI_TRUE—Perform ID query
 VI_FALSE—Skip ID query

Default Value: VI_TRUE

resetDevice ViBoolean Specify whether to reset the device during the initialization process.
 Default Value: VI_TRUE

Defined Values

VI_TRUE (1)—Reset device

VI_FALSE (0)—Do not reset device

optionString Specifies initialization commands. The following table lists the attributes and the name you use in the optionString to identify the attribute.

Attribute Name	Attribute Values
RangeCheck - NISCOPE_ATTR_RANGE_CHECK	VI_TRUE, VI_FALSE
QueryInstrStatus - NISCOPE_QUERY_INSTRUMENT_STATUS	VI_TRUE, VI_FALSE
Cache - NISCOPE_ATTR_CACHE	VI_TRUE, VI_FALSE
Simulate - NISCOPE_ATTR_SIMULATE	VI_TRUE, VI_FALSE

Default Values:

"Simulate=0,RangeCheck=1,QueryInstrStatus=1,Cache=1"

You can use the option string to simulate a device. The DriverSetup flag specifies the model that is to be simulated and the type of the model. One example to simulate a PXI-5105 would be as follows:

```
Option String: Simulate = 1, DriverSetup = Model:5105
; BoardType:PXI
```

Refer to the example `niScope EX Simulated Acquisition` for more information on simulation.

You can also use the option string to attach an accessory such as the PXI-5900 to your digitizer session to allow the seamless use of the accessory:

Option String: `DriverSetup = Accessory:Dev1`

Refer to the example `niScope_EX_External_Amplifier` for more information.

Output

Name	Type	Description
<code>vi</code>	<code>ViSession*</code>	Returns a session handle that you can use to identify the device in all subsequent NI-SCOPE function calls.

Return Value

Name	Type	Description
Status	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_close

C Function Prototype

```
ViStatus niScope_close (ViSession vi);
```

Purpose

When you are finished using an instrument driver session, you must call this function to perform the following actions:

- Closes the instrument I/O session.
- Destroys the IVI session and all of its attributes.
- Deallocates any memory resources used by the IVI session.

Related topics:

- [NI-SCOPE Programming Flow](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_AutoSetup

C Function Prototype

```
ViStatus niScope_AutoSetup (ViSession vi);
```

Purpose

Automatically configures the instrument. When you call this function, the digitizer senses the input signal and automatically configures many of the instrument settings. If a signal is detected on a channel, the driver chooses the smallest available vertical range that is larger than the signal range. For example, if the signal is a 1.2 V_{pk-pk} sine wave, and the device supports 1 V and 2 V vertical ranges, the driver will choose the 2 V vertical range for that channel.

If no signal is found on any analog input channel, a warning is returned, and all channels are enabled. A channel is considered to have a signal present if the signal is at least 10% of the smallest vertical range available for that channel.

The following settings are changed:

General	
Acquisition mode	Normal
Reference clock	Internal
Vertical	
Vertical coupling	AC (when AC is supported; otherwise DC)
Vertical bandwidth	Full
Vertical range	Changed by auto setup
Vertical offset	0 V
Probe attenuation	Unchanged by auto setup
Input impedance	Unchanged by auto setup
Horizontal	
Sample rate	Changed by auto setup
Min record length	Changed by auto setup
Enforce realtime	True
Number of Records	Changed to 1
Triggering	
Trigger type	Edge if signal present, otherwise immediate
Trigger channel	Lowest numbered channel with a signal present
Trigger slope	Positive
Trigger coupling	DC
Reference position	50%
Trigger level	50% of signal on trigger channel
Trigger delay	0
Trigger holdoff	0
Trigger output	None

Related topics:

- [Configuring an Acquisition Using Auto_Setup](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureAcquisition

C Function Prototype

```
ViStatus niScope_ConfigureAcquisition(ViSession vi, ViInt32 acquisitionType);
```

Purpose

Configures how the oscilloscope acquires data and fills the waveform record.

Related topics:

- [Acquisition Functions](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
acquisitionType	ViInt32	Specifies the manner in which the digitizer acquires data and fills the waveform record; NI-SCOPE sets NISCOPE_ATTR_ACQUISITION_TYPE to this value. Defined Values
	<code>NISCOPE_VAL_NORMAL (0)</code>	Sets the oscilloscope to normal resolution mode. The oscilloscope can use real-time sampling or equivalent-time sampling.
	<code>NISCOPE_VAL_FLEXRES (1001)</code>	Sets legacy oscilloscopes to flexible resolution mode, if supported.
	<code>NISCOPE_VAL_DDC (1002)</code>	Sets legacy oscilloscopes to DDC mode, if supported. To use DDC mode for the PXI/PCI-5142 or PXIe-5622, set this control to <code>NISCOPE_VAL_NORMAL</code> and set the NISCOPE_ATTR_DDC_ENABLED attribute to <code>VI_TRUE</code> .

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureHorizontalTiming

C Function Prototype

```
ViStatus niScope_ConfigureHorizontalTiming (ViSession vi, ViReal64
minSampleRate, ViInt32 minNumPts, ViReal64 refPosition, ViInt32 numRecords,
ViBoolean enforceRealtime);
```

Purpose

Configures the common properties of the horizontal subsystem for a multirecord acquisition in terms of minimum sample rate.

Related topics:

- [Configuring the Horizontal Settings](#)
- [NI-TClk Overview](#)
- [Coercions of Horizontal Parameters](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
minSampleRate	ViReal64	The sampling rate for the acquisition. Refer to NISCOPE_ATTR_MIN_SAMPLE_RATE for more information.
minNumPts	ViInt32	The minimum number of points you need in the record for each channel; call niScope_ActualRecordLength to obtain the actual record length used. Valid Values: Greater than 1; limited by available memory
refPosition	ViReal64	The position of the Reference Event in the waveform record specified as a percentage.
numRecords	ViInt32	The number of records to acquire
enforceRealtime	ViBoolean	Indicates whether the digitizer enforces real-time measurements or allows equivalent-time (RIS) measurements; not all digitizers support RIS—refer to Features Supported by Device for more information.

Default value: VI_TRUE

Defined Values

VI_TRUE—Allow real-time acquisitions only

VI_FALSE—Allow real-time and equivalent-time acquisitions

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureChanCharacteristics

C Function Prototype

```
ViStatus niScope_ConfigureChanCharacteristics (ViSession vi, ViConstString channelList, ViReal64 inputImpedance, ViReal64 maxInputFrequency);
```

Purpose

Configures the attributes that control the electrical characteristics of the channel—the input impedance and the bandwidth.

Related topics:

- [Using Configure Chan Characteristics](#)
- [Impedance and Impedance Matching](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
inputImpedance	ViReal64	The input impedance for the channel; NI-SCOPE sets NISCOPE_ATTR_INPUT_IMPEDANCE to this value.
maxInputFrequency	ViReal64	The bandwidth for the channel; NI-SCOPE sets NISCOPE_ATTR_MAX_INPUT_FREQUENCY to this value. Pass 0 for this value to use the hardware default bandwidth. Pass -1 for this value to achieve full bandwidth.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureVertical

C Function Prototype

```
ViStatus niScope_ConfigureVertical (ViSession vi, ViConstString channelList,
ViReal64 range, ViReal64 offset, ViInt32 coupling, ViReal64 probeAttenuation,
ViBoolean enabled);
```

Purpose

Configures the most commonly configured attributes of the digitizer vertical subsystem, such as the range, offset, coupling, probe attenuation, and the channel.

Related topics:

- [Configuring the Vertical Settings](#)
- [NI-SCOPE Programming Flow](#)
- [Coercions of Vertical Parameters](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
range	ViReal64	Specifies the vertical range. Refer to NISCOPE_ATTR_VERTICAL_RANGE for more information.
offset	ViReal64	Specifies the vertical offset. Refer to NISCOPE_ATTR_VERTICAL_OFFSET for more information.
coupling	ViInt32	Specifies how to couple the input signal. Refer to NISCOPE_ATTR_VERTICAL_COUPLING for more information.
probeAttenuation	ViReal64	Specifies the probe attenuation. Refer to NISCOPE_ATTR_PROBE_ATTENUATION for valid values.
enabled	ViBoolean	Specifies whether the channel is enabled for acquisition. Refer to NISCOPE_ATTR_CHANNEL_ENABLED for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ActualMeasWfmSize

C Function Prototype

```
ViStatus niScope_ActualMeasWfmSize(ViSession vi, ViInt32 arrayMeasFunction, ViInt32* measWfmSize);
```

Purpose

Returns the total available size of an array measurement acquisition.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
arrayMeasFunction	ViInt32	The array measurement to perform.

Output

Name	Type	Description
measWfmSize	ViInt32*	Returns the size (in number of samples) of the resulting analysis waveform.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ActualNumWfms

C Function Prototype

```
ViStatus niScope_ActualNumWfms (ViSession vi, ViConstString channelList, ViInt32* numWfms);
```

Purpose

Helps you to declare appropriately sized waveforms. NI-SCOPE handles the channel list parsing for you.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .

Output

Name	Type	Description
numWfms	ViInt32*	Returns the number of records times the number of channels; if the niScope_ConfigureAcquisition <code>acquisitionType</code> is set to <code>NISCOPE_VAL_DDC</code> , this value is multiplied by two.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ActualRecordLength

C Function Prototype

```
ViStatus niScope_ActualRecordLength (ViSession vi, ViInt32* recordLength);
```

Purpose

Returns the actual number of points the digitizer acquires for each channel. After configuring the digitizer for an acquisition, call this function to determine the size of the waveforms that the digitizer acquires. The value is equal to or greater than the minimum number of points specified in any of the Configure Horizontal functions.

Related topics:

- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Output

Name	Type	Description
recordLength	ViInt32*	Returns the actual number of points the digitizer acquires for each channel; NI-SCOPE returns the value held in the NISCOPE_ATTR_HORZ_RECORD_LENGTH attribute.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SampleMode

C Function Prototype

```
ViStatus niScope_SampleMode (ViSession vi, ViInt32* sampleMode);
```

Purpose

Returns the sample mode the digitizer is currently using.

Related topics:

- [Sampling Methods](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
Output		
Name	Type	Description
sampleMode	ViInt32*	Returns the sample mode the digitizer is currently using; NI-SCOPE returns the value of the NISCOPE_ATTR_SAMPLE_MODE attribute.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_SampleRate

C Function Prototype

```
ViStatus niScope_SampleRate (ViSession vi, ViReal64* actualSampleRate);
```

Purpose

Returns the effective sample rate, in samples per second, of the acquired waveform using the current configuration. Refer to [Coercions of Horizontal Parameters](#) for more information about sample rate coercion.

Related topics:

- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)
- [Sample Clock](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
Output		

Name	Type	Description
actualSampleRate	ViReal64*	Returns the effective sample rate of the acquired waveform the digitizer acquires for each channel; the driver returns the value held in the <code>NISCOPE_ATTR_HORZ_SAMPLE_RATE</code> attribute.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerDigital

C Function Prototype

```
ViStatus niScope_ConfigureTriggerDigital (ViSession vi, ViConstString triggerSource, ViInt32 slope, ViReal64 holdoff, ViReal64 delay);
```

Purpose

Configures the common properties of a digital trigger.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the `NISCOPE_ATTR_ACO_ARM_SOURCE` (Start Trigger Source) attribute. The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.



Notes For multirecord acquisitions, all records after the first record are started by using the Advance Trigger Source. The default is immediate.

You can adjust the amount of pre-trigger and post-trigger samples using the reference position parameter on the [niScope_ConfigureHorizontalTiming](#) function. The default is half of the record length.

Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Digital triggering is not supported in RIS mode.

Related topics:

- [Digital Triggers](#)
- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to NISCOPE_ATTR_TRIGGER_SOURCE for defined values.
slope	ViInt32	Specifies whether you want a rising edge or a falling edge to trigger the digitizer. Refer to NISCOPE_ATTR_TRIGGER_SLOPE for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerEdge

C Function Prototype

```
ViStatus niScope_ConfigureTriggerEdge (ViSession vi, ViConstString triggerSource,
ViReal64 level, ViInt32 slope, ViInt32 triggerCoupling, ViReal64 holdoff, ViReal64
delay);
```

Purpose

Configures common properties for analog edge triggering.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACQ_ARM_SOURCE](#) (Start Trigger Source) attribute. The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Edge Triggers](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to NISCOPE_ATTR_TRIGGER_SOURCE for defined values.
level	ViReal64	The voltage threshold for the trigger. Refer to NISCOPE_ATTR_TRIGGER_LEVEL for more information.
slope	ViInt32	Specifies whether you want a rising edge or a falling edge to trigger the digitizer. Refer to NISCOPE_ATTR_TRIGGER_SLOPE for more information.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to NISCOPE_ATTR_TRIGGER_COUPLING for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerGlitch

ViStatus niScope_ConfigureTriggerGlitch (ViSession Instrument_Handle, ViChar _VI_FAR Trigger_Source[], ViReal64 Level, ViReal64 Width, ViInt32 Polarity, ViInt32 Condition, ViInt32 Trigger_Coupling, ViReal64 Holdoff, ViReal64 Delay);

Purpose

Configures common properties for glitch triggering.

A glitch trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a width that is either greater than or less than a duration you specify.

Trigger System Operation

When you initiate an acquisition, the trigger system operates in the following manner.

- The digitizer waits for the start trigger, which is configured through the `NISCOPE_ATTR_ACQ_ARM_SOURCE` attribute. The default is `NISCOPE_VAL_IMMEDIATE`.
- Upon receiving the start trigger, the digitizer begins sampling pretrigger points.
- After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a trigger function such as this one.
- Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling.

With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay. Additionally, you can adjust the amount of pretrigger and posttrigger samples using the `refPosition` input of the `niScope_ConfigureHorizontalTiming` function. The default is half the record length.

For multirecord acquisitions, all records after the first record are started based on the setting of the `NISCOPE_ATTR_ADV_TRIG_SRC` attribute. The default value is `NISCOPE_VAL_IMMEDIATE`.



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Glitch Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to <code>NISCOPE_ATTR_TRIGGER_SOURCE</code> for defined values.
level	ViReal64	The voltage threshold for the trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_LEVEL</code> for more information.
Width	ViReal64	Specifies, in seconds, the glitch duration to be used in combination with the glitch condition that triggers the oscilloscope. The oscilloscope triggers when it detects a pulse of duration either less than or greater than this value depending on the value of the <code>NISCOPE_ATTR_GLITCH_CONDITION</code> attribute. Refer to the <code>NISCOPE_ATTR_GLITCH_WIDTH</code> attribute for more information.
Polarity	ViInt32	Specifies the polarity of the pulses that trigger the oscilloscope for glitch triggering. Refer to the <code>NISCOPE_ATTR_GLITCH_POLARITY</code> attribute for defined values.
Glitch Condition	ViInt32	Specifies whether the oscilloscope triggers on pulses of duration less than or greater than the specified <code>NISCOPE_ATTR_GLITCH_WIDTH</code> . Refer to the <code>NISCOPE_ATTR_GLITCH_CONDITION</code> attribute for defined values.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to <code>NISCOPE_ATTR_TRIGGER_COUPLING</code> for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_HOLDOFF</code> for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to <code>NISCOPE_ATTR_TRIGGER_DELAY_TIME</code> for more information.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureTriggerHysteresis


C Function Prototype

```
ViStatus niScope_ConfigureTriggerHysteresis (ViSession vi, ViConstString
triggerSource, ViReal64 level, ViReal64 hysteresis, ViInt32 slope, ViInt32
triggerCoupling, ViReal64 holdoff, ViReal64 delay);
```

Purpose

Configures common properties for analog hysteresis triggering. This kind of trigger specifies an additional value, specified in the hysteresis parameter, that a signal must pass through before a trigger can occur. This additional value acts as a kind of buffer zone that keeps noise from triggering an acquisition.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACO_ARM_SOURCE \(Start Trigger Source\)](#). The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.

 **Note** Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Hysteresis Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to <code>NISCOPE_ATTR_TRIGGER_SOURCE</code> for defined values.
level	ViReal64	The voltage threshold for the trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_LEVEL</code> for more information.
hysteresis	ViReal64	The size of the hysteresis window on either side of the level in volts; the digitizer triggers when the trigger signal passes through the hysteresis value you specify with this parameter, has the slope you specify with <code>slope</code> , and passes through the level. Refer to <code>NISCOPE_ATTR_TRIGGER_HYSTERESIS</code> for defined values.
slope	ViInt32	Specifies whether you want a rising edge or a falling edge to trigger the digitizer. Refer to <code>NISCOPE_ATTR_TRIGGER_SLOPE</code> for more information.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to <code>NISCOPE_ATTR_TRIGGER_COUPLING</code> for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_HOLDOFF</code> for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to <code>NISCOPE_ATTR_TRIGGER_DELAY_TIME</code> for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information

concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerImmediate

C Function Prototype

```
ViStatus niScope_ConfigureTriggerImmediate (ViSession vi);
```

Purpose

Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.

When you initiate an acquisition, the digitizer waits for a trigger. You specify the type of trigger that the digitizer waits for with a Configure Trigger function, such as `niScope_ConfigureTriggerImmediate`.

Related topics:

- [Immediate Triggers](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information

concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerRunt

ViStatus niScope_ConfigureTriggerRunt (ViSession Instrument_Handle, ViChar _VI_FAR Trigger_Source[], ViReal64 Low_Threshold, ViReal64 High_Threshold, ViInt32 Polarity, ViInt32 Trigger_Coupling, ViReal64 Holdoff, ViReal64 Delay);

Purpose

Configures common properties for runt triggering.

A runt trigger occurs when both the leading edge and trailing edge of a pulse cross only one of two trigger thresholds you specify and with a polarity you specify, where the polarity is relative to the threshold crossed.

Trigger System Operation

When you initiate an acquisition, the trigger system operates in the following manner.

- The digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACQ_ARM_SOURCE](#) attribute. The default is `NISCOPE_VAL_IMMEDIATE`.
- Upon receiving the start trigger, the digitizer begins sampling pretrigger points.
- After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a trigger function such as this one.
- Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling.

With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay. Additionally, you can adjust the amount of pretrigger and posttrigger samples using the refPosition input of the [niScope_ConfigureHorizontalTiming](#) function. The default is half the record length.

For multirecord acquisitions, all records after the first record are started based on the setting of the [NISCOPE_ATTR_ADV_TRIG_SRC](#) attribute. The default value is `NISCOPE_VAL_IMMEDIATE`.



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Runt Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to NISCOPE_ATTR_TRIGGER_SOURCE for defined values.
Low Threshold	ViReal64	Specifies, in volts, the lower of two thresholds that bound the vertical range to examine for runt pulses. Refer to the NISCOPE_ATTR_RUNT_LOW_THRESHOLD attribute for more information.
High Threshold	ViReal64	Specifies, in volts, the higher of two thresholds that bound the vertical range to examine for runt pulses. Refer to the NISCOPE_ATTR_RUNT_HIGH_THRESHOLD attribute for more information.
Polarity	ViInt32	Specifies the polarity of the runt pulses, relative to the runt threshold the pulses cross, that trigger the oscilloscope for runt triggering. Refer to the NISCOPE_ATTR_RUNT_POLARITY attribute for defined values.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to NISCOPE_ATTR_TRIGGER_COUPLING for more information.

holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISC OPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISC OPE_ATTR_TRIGGER_DELAY_TI ME for more information.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureTriggerSoftware

C Function Prototype

ViStatus niScope_ConfigureTriggerSoftware (ViSession vi, ViReal64 holdoff, ViReal64 delay);

Purpose

Configures common properties for software triggering.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [NISC OPE_ATTR_ACQ_ARM_SOURCE](#) (Start Trigger Source) attribute. The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure

Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.

To trigger the acquisition, use [niScope_SendSoftwareTriggerEdge](#).



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Software Triggers](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerVideo

C Function Prototype

```
ViStatus niScope_ConfigureTriggerVideo (ViSession vi, ViConstString triggerSource,
ViBoolean enableDCRestore, ViInt32 signalFormat, ViInt32 event, ViInt32
lineNumber, ViInt32 polarity, ViInt32 triggerCoupling, ViReal64 holdoff, ViReal64
delay);
```

Purpose

Configures the common properties for video triggering, including the signal format, TV event, line number, polarity, and enable DC restore. A video trigger occurs when the digitizer finds a valid video signal sync.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACO_ARM_SOURCE](#) (Start Trigger Source) attribute. The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to NISCOPE_ATTR_TRIGGER_SOURCE for defined values.

enableDCRestore	ViBoolean	Offsets each video line so the clamping level (the portion of the video line between the end of the color burst and the beginning of the active image) is moved to zero volt. Refer to NISCOPE_ATTR_ENABLE_DC_RESTORE for defined values.
signalFormat	ViInt32	Specifies the type of video signal sync the digitizer should look for. Refer to NISCOPE_ATTR_TV_TRIGGER_SIGNAL_FORMAT for more information.
event	ViInt32	Specifies the TV event you want to trigger on. You can trigger on a specific or on the next coming line or field of the signal.
lineNumber	ViInt32	Selects the line number to trigger on. The line number range covers an entire frame and is referenced as shown on Vertical Blanking and Synchronization Signal . Refer to NISCOPE_ATTR_TV_TRIGGER_LINE_NUMBER for more information. Default value: 1
polarity	ViInt32	Specifies the polarity of the video signal sync.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to NISCOPE_ATTR_TRIGGER_COUPLING for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureTriggerWidth

ViStatus niScope_ConfigureTriggerWidth (ViSession Instrument_Handle, ViChar _VI_FAR Trigger_Source[], ViReal64 Level, ViReal64 Low_Threshold, ViReal64 High_Threshold, ViInt32 Polarity, ViInt32 Condition, ViInt32 Trigger_Coupling, ViReal64 Holdoff, ViReal64 Delay);

Purpose

Configures common properties for width triggering.

A width trigger occurs when a pulse that crosses a vertical threshold you specify and with a polarity you specify also has a duration that is either within or outside a duration range you specify.


Trigger System Operation

When you initiate an acquisition, the trigger system operates in the following manner.

- The digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACO_ARM_SOURCE](#) attribute. The default is `NISCOPE_VAL_IMMEDIATE`.
- Upon receiving the start trigger, the digitizer begins sampling pretrigger points.
- After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a trigger function such as this one.
- Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling.

With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay. Additionally, you can adjust the amount of pretrigger and posttrigger samples using the `refPosition` input of the [niScope_ConfigureHorizontalTiming](#) function. The default is half the record length.

For multirecord acquisitions, all records after the first record are started based on the setting of the `NISCOPE_ATTR_ADV_TRIG_SRC` attribute. The default value is `NISCOPE_VAL_IMMEDIATE`.

 **Note** Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Width Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to <code>NISCOPE_ATTR_TRIGGER_SOURCE</code> for defined values.
level	ViReal64	The voltage threshold for the trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_LEVEL</code> for more information.
Low Threshold	ViReal64	Specifies, in seconds, the lower bound on the range of pulse durations that triggers the oscilloscope. Refer to the <code>NISCOPE_ATTR_TRIGGER_WIDTH_LOW_THRESHOLD</code> attribute for defined values.
High Threshold	ViReal64	Specifies, in seconds, the upper bound on the range of pulse durations that triggers the oscilloscope. Refer to the <code>NISCOPE_ATTR_TRIGGER_WIDTH_HIGH_THRESHOLD</code> attribute for defined values.
Polarity	ViInt32	Specifies the polarity of the pulses that trigger the oscilloscope for width triggering. Refer to the <code>NISCOPE_ATTR_TRIGGER_WIDTH_POLARITY</code> attribute for defined values.
Condition	ViInt32	Specifies whether the oscilloscope triggers on pulses of duration within or outside the range of pulse durations bounded by Low Threshold and High Threshold. Refer to the <code>NISCOPE_ATTR_TRIGGER_WIDTH_CONDITION</code> attribute for defined values.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to <code>NISCOPE_ATTR_TRIGGER_COUPLING</code> for more information.

holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ConfigureTriggerWindow

C Function Prototype

```
ViStatus niScope_ConfigureTriggerWindow (ViSession vi, ViConstString triggerSource, ViReal64 lowLevel, ViReal64 highLevel, ViInt32 windowMode, ViInt32 triggerCoupling, ViReal64 holdoff, ViReal64 delay);
```

Purpose

Configures common properties for analog window triggering. A window trigger occurs when a signal enters or leaves a window you specify with the high level or low level parameters.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [NISCOPE_ATTR_ACQ_ARM_SOURCE](#) (Start Trigger Source) attribute. The default is immediate. Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling

pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a function such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger function, you specify configuration parameters such as the trigger source and the amount of trigger delay.

To trigger the acquisition, use [niScope_SendSoftwareTriggerEdge](#).



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Window Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerSource	ViConstString	Specifies the trigger source. Refer to NISCOPE_ATTR_TRIGGER_SOURCE for defined values.
lowLevel	ViReal64	Passes the voltage threshold you want the digitizer to use for low triggering.
highLevel	ViReal64	Passes the voltage threshold you want the digitizer to use for high triggering.
windowMode	ViInt32	Specifies whether you want the trigger to occur when the signal enters or leaves a window.
triggerCoupling	ViInt32	Applies coupling and filtering options to the trigger signal. Refer to NISCOPE_ATTR_TRIGGER_COUPLING for more information.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to NISCOPE_ATTR_TRIGGER_HOLDOFF for more information.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to NISCOPE_ATTR_TRIGGER_DELAY_TIME for more information.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_SendSoftwareTriggerEdge

C Function Prototype

```
ViStatus niScope_SendSoftwareTriggerEdge (ViSession vi, ViInt32 whichTrigger);
```

Purpose

Sends the selected trigger to the digitizer. Call this function if you called [niScope_ConfigureTriggerSoftware](#) when you want the Reference trigger to occur. You can also call this function to override a misused edge, digital, or hysteresis trigger. If you have configured [NISCOPE_ATTR_Acq_ARM_SOURCE](#), [NISCOPE_ATTR_ARM_REF_TRIG_SRC](#), or [NISCOPE_ATTR_ADV_TRIG_SRC](#), call this function when you want to send the corresponding trigger to the digitizer.

Related topics:

- [Software Triggers](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

whichTrigger ViInt32 Specifies the type of trigger to send to the digitizer.

Defined Values

```
NISCOPE_VAL_SOFTWARE_TRIGGER_START (0L)
NISCOPE_VAL_SOFTWARE_TRIGGER_ARM_REFERENCE (1L)
NISCOPE_VAL_SOFTWARE_TRIGGER_REFERENCE (2L)
NISCOPE_VAL_SOFTWARE_TRIGGER_ADVANCE (3L)
```

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_AdjustSampleClockRelativeDelay

C Function Prototype

```
ViStatus niScope_AdjustSampleClockRelativeDelay(ViSession vi, ViReal64 delay);
```

Purpose

Applies offset, in seconds, to the sample clock relative to the reference clock when using the onboard clock.

Each time this function is called, the sample clock is offset from the reference clock by the specified amount of time.

Adjustment range: ± 1 **Sample Clock Period**

Programming Patterns

Use this function when performing manual adjustment to correct for skew and jitter between oscilloscopes synchronized with NI-TClk.

1. Apply time offset with this function to reduce skew and/or jitter.
2. Use the `NISCOPE_ATTR_OSCILLATOR_PHASE_DAC_VALUE` attribute to convert this time offset into a value that can apply the manual adjustment across sessions and improve synchronization repeatability.

For details on performing manual adjustment, refer to [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#).

Related topics:

- [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#)
- [Sample Clock](#)
- [Reference Clock/Phase-Lock Loop](#)
- [NI-TClk Overview](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
delay	ViReal64	How long the digitizer waits after receiving the trigger to start acquiring data. Refer to <code>NISCOPE_ATTR_TRIGGER_DELAY_TIME</code> for more information.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
-------	---------

0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureClock

C Function Prototype

```
ViStatus niScope_ConfigureClock (ViSession vi, ViConstString inputClockSource,
ViConstString outputClockSource, ViConstString clockSyncPulseSource, ViBoolean
masterEnabled);
```

Purpose

Configures the attributes for synchronizing the digitizer to a reference or sending the digitizer's reference clock output to be used as a synchronizing clock for other digitizers.



Note Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Sample Clock](#)
- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
inputClockSource	ViConstString	Specifies the input source for the PLL reference clock (such as the 1-20 MHz clock on SMC-based devices) to which the digitizer is phase-locked for all digitizers. Refer to NISCOP_E_ATTR_INPUT_CLOCK_SOURCE for more information.
outputClockSource	ViConstString	Specifies the output source for the reference clock to which the sample clock of another oscilloscope can be phase-

		locked. Refer to NISCOPE_ATTR_OUTPUT_CLOCK_SOURCE for more information
clockSyncPulseSource	ViConstString	Specifies the line on which the sample clock or the one-time sync pulse is sent or received. This line should be the same for all devices to be synchronized. Refer to NISCOPE_ATTR_CLOCK_SYNC_PULSE_SOURCE for more information.
masterEnabled	ViBoolean	Specifies whether the device is a master or a slave. The master device is typically the originator of the trigger signal and the clock sync pulse. For a standalone device, set this control to VI_FALSE.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ExportSignal

C Function Prototype

ViStatus niScope_ExportSignal (ViSession vi, ViInt32 signal, ViConstString signalIdentifier, ViConstString outputTerminal);

Purpose

 **Note** This function replaces [niScope_ConfigureTriggerOutput](#).

Configures the digitizer to generate a signal that other devices can detect when configured for digital triggering or sharing clocks. The signal parameter specifies what condition causes the digitizer to generate the signal. The outputTerminal

parameter specifies where to send the signal on the hardware (such as a PFI connector or RTSI line).

In cases where multiple instances of a particular signal exist, use the `signalIdentifier` input to specify which instance to control. For normal signals, only one instance exists and you should leave this parameter set to the empty string. You can call this function multiple times and set each available line to a different signal.

To unprogram a specific line on device, call this function with the signal you no longer want to export and set `outputTerminal` to `NISCOPE_VAL_NONE`.

Related topics:

- [Triggering](#)
- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [PXI Trigger Lines](#)

Parameters

Input											
Name	Type	Description									
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.									
<code>signal</code>	<code>ViInt32</code>	Signal (clock, trigger, or event) to export. Defined Values									
		<table border="1"> <tbody> <tr> <td><code>NISCOPE_VAL_REF_TRIGGER</code></td> <td>(1)</td> <td>Generate a pulse when detecting the Stop/Reference trigger.</td> </tr> <tr> <td><code>NISCOPE_VAL_START_TRIGGER</code></td> <td>(2)</td> <td>Generate a pulse when detecting a Start trigger.</td> </tr> <tr> <td><code>NISCOPE_VAL_END_OF_ACQUISITION_EVENT</code></td> <td>(3)</td> <td>Generate a pulse when the acquisition finishes.</td> </tr> </tbody> </table>	<code>NISCOPE_VAL_REF_TRIGGER</code>	(1)	Generate a pulse when detecting the Stop/Reference trigger.	<code>NISCOPE_VAL_START_TRIGGER</code>	(2)	Generate a pulse when detecting a Start trigger.	<code>NISCOPE_VAL_END_OF_ACQUISITION_EVENT</code>	(3)	Generate a pulse when the acquisition finishes.
<code>NISCOPE_VAL_REF_TRIGGER</code>	(1)	Generate a pulse when detecting the Stop/Reference trigger.									
<code>NISCOPE_VAL_START_TRIGGER</code>	(2)	Generate a pulse when detecting a Start trigger.									
<code>NISCOPE_VAL_END_OF_ACQUISITION_EVENT</code>	(3)	Generate a pulse when the acquisition finishes.									

NISCOPE_VAL_END_OF_RECORD_EVENT	(4)	Generate a pulse at the end of the record.
NISCOPE_VAL_ADVANCE_TRIGGER	(5)	Generate a pulse when detecting an Advance trigger.
NISCOPE_VAL_READY_FOR_ADVANCE_EVENT	(6)	Asserts when the digitizer is ready to advance to the next record.
NISCOPE_VAL_READY_FOR_START_EVENT	(7)	Asserts when the digitizer is initiated and ready to accept a Start trigger and begin sampling.
NISCOPE_VAL_READY_FOR_REF_EVENT	(10)	Asserts when the digitizer is ready to accept a Reference trigger.
NISCOPE_VAL_REF_CLOCK	(100)	Export the Reference clock for the digitizer to the specified terminal.
NISCOPE_VAL_SAMPLE_CLOCK	(101)	Export the Sample clock for the digitizer to the specified terminal.
NISCOPE_VAL_5V_OUT	(13)	Exports a 5 V power supply.

signalIdentifier ViConstString Describes the signal being exported.

outputTerminal ViConstString Identifies the hardware signal line on which the digital pulse is generated.

Defined Values

NISCOPE_VAL_RTISI_0	("VAL_RTISI_0")
NISCOPE_VAL_RTISI_1	("VAL_RTISI_1")
NISCOPE_VAL_RTISI_2	("VAL_RTISI_2")
NISCOPE_VAL_RTISI_3	("VAL_RTISI_3")
NISCOPE_VAL_RTISI_4	("VAL_RTISI_4")
NISCOPE_VAL_RTISI_5	("VAL_RTISI_5")
NISCOPE_VAL_RTISI_6	("VAL_RTISI_6")
NISCOPE_VAL_RTISI_7	("VAL_RTISI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_CLK_OUT	("VAL_CLK_OUT")

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetEqualizationFilterCoefficients

C Function Prototype

ViStatus _VI_FUNC niScope_GetEqualizationFilterCoefficients (ViSession vi, ViConstString channel, ViInt32 numberOfCoefficients, ViReal64* coefficients);

Purpose

Retrieves the custom coefficients for the equalization FIR filter on the device. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
numberOfCoefficients	ViInt32	The number of coefficients being passed in the coefficients array.
Output		
Name	Type	Description
coefficients	ViReal64*	The custom coefficients for the equalization FIR filter on the device. These coefficients should be between +1 and -1. You can obtain the number of coefficients from the NISCOPE_ATTR_EQUALIZATION_NUM_COEFFICIENTS attribute.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetFrequencyResponse

```
ViStatus _VI_FUNC niScope_GetFrequencyResponse (ViSession vi, ViConstString
channelName, ViInt32 bufferSize, ViReal64 frequencies[], ViReal64 amplitudes[],
ViReal64 phases[], ViInt32* numberOfFrequencies);
```

Purpose

Gets the frequency response of the digitizer for the current configurations of the channel attributes. Not all digitizers support this function.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
bufferSize	ViInt32	The array size for the frequencies, amplitudes, and phases arrays that you pass in to the other parameters. To determine the sizes of the buffers to allocate for the frequencies, amplitudes, and phases arrays, pass a value of 0 to the bufferSize parameter and a value of NULL to the frequencies parameter. In this case, the value returned by the numberOfFrequencies parameter is the size of the arrays necessary to hold the frequencies, amplitudes, and phases. Allocate three arrays of this size, then call this function again (with correct bufferSize parameter) to retrieve the actual values.
Output		
Name	Type	Description
frequencies	ViReal64	The array of frequencies that corresponds with the amplitude and phase response of the device.
amplitudes	ViReal64	The array of amplitudes that correspond with the magnitude response of the device.

phases	ViReal64	The array of phases that correspond with the phase response of the device.
numberOfFrequencies	ViInt32	Returns the number of frequencies in the returned spectrum.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureEqualizationFilterCoefficients

C Function Prototype

```
ViStatus _VI_FUNC niScope_ConfigureEqualizationFilterCoefficients (ViSession vi, ViConstString channel, ViInt32 numberOfCoefficients, ViReal64* coefficients);
```

Purpose

Configures the custom coefficients for the equalization FIR filter on the device. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
numberOfCoefficients	ViInt32	The number of coefficients being passed in the coefficients array.
coefficients	ViReal64*	The custom coefficients for the equalization FIR filter on the device. These coefficients should be between +1 and -1. You can obtain the number of coefficients from the NISCOP E_ATTR_EQUALIZATION_NUM_COEFFICIENTS attribute. The NISCOPE_ATTR_EQUALIZATION_FILTER_ENABLED attribute must be set to TRUE to enable the filter.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ImportAttributeConfigurationFile

ViStatus niScope_ImportAttributeConfigurationFile (ViSession vi, ViConstString filePath);

Purpose

Imports an attribute configuration to the session from the specified file.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels..



Note You cannot call this function while the session is in a running state, such as while acquiring a signal.

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the `resourceName` input to the `niScope_InitWithOptions` or `niScope_init` functions.

For example, if your entry for `resourceName` is `PXI1Slot1`, `PXI1Slot2` for the exporting session and `PXI2Slot2`, `PXI2Slot3` for the importing session:

- The configuration exported from `PXI1Slot1` is imported into `PXI2Slot2`.
- The configuration exported from `PXI1Slot2` is imported into `PXI2Slot3`.



Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

Related Topics:

[Attributes and Attribute Functions](#)

[Setting Attributes Before Reading Attributes](#)

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>filePath</code>	<code>ViConstString</code>	The absolute path to the file that contains the attribute configuration to import. If you specify an empty or relative path, this function returns an error.
		Default File Extension: <code>.niscopesconfig</code>

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ExportAttributeConfigurationFile

ViStatus niScope_ExportAttributeConfigurationFile (ViSession vi, ViConstString filePath);

Purpose

Exports the attribute configuration of the session to the specified file.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

This function verifies that the attributes you have configured for the session are valid. If the configuration is invalid, NI-SCOPE returns an error.

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the `resourceName` input to the [niScope_InitWithOptions](#) or [niScope_init](#) functions.

For example, if your entry for `resourceName` is `PXI1Slot1`, `PXI1Slot2` for the exporting session and `PXI2Slot2`, `PXI2Slot3` for the importing session:

- The configuration exported from `PXI1Slot1` is imported into `PXI2Slot2`.
- The configuration exported from `PXI1Slot2` is imported into `PXI2Slot3`.



Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

Related Topics:

[Attributes and Attribute Functions](#)

[Setting Attributes Before Reading Attributes](#)

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>filePath</code>	<code>ViConstString</code>	The absolute path to a placeholder file you must create to contain the attribute configuration you want to export. If you specify an empty or relative path, this function returns an error. Default file extension: <code>.niscscopeconfig</code>

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values

Errors

niScope_ImportAttributeConfigurationBuffer

ViStatus niScope_ImportAttributeConfigurationBuffer (ViSession vi, ViInt32 size, ViAddr configuration);

Purpose

Imports an attribute configuration to the session from the specified buffer.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.



Note You cannot call this function while the session is in a running state, such as while acquiring a signal.

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the `resourceName` input to the [niScope_InitWithOptions](#) or [niScope_init](#) functions.

For example, if your entry for `resourceName` is `PXI1Slot1, PXI1Slot2` for the exporting session and `PXI2Slot2, PXI2Slot3` for the importing session:

- The configuration exported from `PXI1Slot1` is imported into `PXI2Slot2`.
- The configuration exported from `PXI1Slot2` is imported into `PXI2Slot3`.



Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

Related Topics:

[Attributes and Attribute Functions](#)

Setting Attributes Before Reading Attributes

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
size	ViInt32	Specifies the size, in bytes, of the byte array to import. If you enter 0, this function returns the needed size.
configuration	ViAddr	Specifies the byte array that contains the attribute configuration to import.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ExportAttributeConfigurationBuffer

ViStatus niScope_ExportAttributeConfigurationBuffer (ViSession vi, ViInt32 size, ViAddr configuration);

Purpose

Exports the attribute configuration of the session to a buffer.

You can export and import session attribute configurations only between NI-SCOPE devices with identical bus types, model numbers, channel counts, and onboard

memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

This function verifies that the attributes you have configured for the session are valid. If the configuration is invalid, NI-SCOPE returns an error.

Device Mapping Behavior

When exporting and importing configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the `resourceName` input to the [niScope_InitWithOptions](#) or [niScope_init](#) functions.

For example, if your entry for `resourceName` is `PXI1Slot1, PXI1Slot2` for the exporting session and `PXI2Slot2, PXI2Slot3` for the importing session:

- The configuration exported from `PXI1Slot1` is imported into `PXI2Slot2`.
- The configuration exported from `PXI1Slot2` is imported into `PXI2Slot3`.



Note NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

Related Topics:

[Attributes and Attribute Functions](#)

[Setting Attributes Before Reading Attributes](#)

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>size</code>	<code>ViInt32</code>	Specifies the size, in bytes, of the byte array to export. If you enter 0, this function returns the needed size.
<code>configuration</code>	<code>ViAddr</code>	Specifies the byte array that contains the exported configuration.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_SetAttributeViInt32

C Function Prototype

```
ViStatus niScope_SetAttributeViInt32 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViInt32 value);
```

Purpose

Sets the value of a ViInt32 attribute. This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level functions as much as possible because they handle order dependencies and multithread locking for you. In addition, high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple attributes perform

instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViInt32	The value that you want to set the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SetAttributeViInt64

C Function Prototype

```
ViStatus niScope_SetAttributeViInt64 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViInt64 value);
```

Purpose

Sets the value of a `ViInt64` attribute. This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level functions as much as possible because they handle order dependencies and multithread locking for you. In addition, high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>channelList</code>	<code>ViConstString</code>	The channel to configure. For more information, refer to Channel String Syntax .
<code>attributeID</code>	<code>ViAttr</code>	The ID of an attribute.
<code>value</code>	<code>ViInt64</code>	The value that you want to set the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SetAttributeViReal64

C Function Prototype

```
ViStatus niScope_SetAttributeViReal64 (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViReal64 value);
```

Purpose

Sets the value of a ViReal64 attribute. This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level driver functions as much as possible because they handle order dependencies and multithread locking for you. In addition, the high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
------	------	-------------

vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViReal64	The value that you want to set the attribute to. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SetAttributeViString

C Function Prototype

```
ViStatus niScope_SetAttributeViString (ViSession vi, ViConstString channelList, ViAttr attributeID, ViConstString value);
```

Purpose

Sets the value of a ViString attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.

- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level driver functions as much as possible because they handle order dependencies and multithread locking for you. In addition, the high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the `SetAttribute` functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>channelList</code>	<code>ViConstString</code>	The channel to configure. For more information, refer to Channel String Syntax .
<code>attributeID</code>	<code>ViAttr</code>	The ID of an attribute.
<code>value</code>	<code>ViConstString</code>	The value that you want to set the attribute to. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SetAttributeViBoolean

C Function Prototype

```
ViStatus niScope_SetAttributeViBoolean (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViBoolean value);
```

Purpose

Sets the value of a ViBoolean attribute. This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level driver functions as much as possible because they handle order dependencies and multithread locking for you. In addition, the high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViBoolean	The value that you want to set the attribute to. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_SetAttributeViSession

C Function Prototype

```
ViStatus niScope_SetAttributeViSession (ViSession vi, ViConstString channelList, ViAttr attributeID, ViSession value);
```

Purpose

Sets the value of a ViSession attribute. This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.



Note NI-SCOPE contains high-level functions that set most of the instrument attributes. Use the high-level driver functions as much as possible because they handle order dependencies and multithread locking for you. In addition, the high-level functions perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the Set Attribute functions, the functions check the instrument status after each call. Also, when state caching is enabled, the high-level functions that configure multiple

attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViSession	The value that you want to set the attribute to. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetAttributeViInt32

C Function Prototype

```
ViStatus niScope_GetAttributeViInt32 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViInt32* value);
```

Purpose

Queries the value of a `ViInt32` attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>channelList</code>	<code>ViConstString</code>	The channel to configure. For more information, refer to Channel String Syntax .
<code>attributeID</code>	<code>ViAttr</code>	The ID of an attribute.
Output		
Name	Type	Description
<code>value</code>	<code>ViInt32*</code>	Returns the current value of the attribute.

Return Value

Name	Type	Description								
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>			Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetAttributeViInt64

C Function Prototype

```
ViStatus niScope_GetAttributeViInt64 (ViSession vi, ViConstString channelList, ViAttr
attributeID, ViInt64* value);
```

Purpose

Queries the value of a ViInt64 attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
Output		
Name	Type	Description
value	ViInt64*	Returns the current value of the attribute.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetAttributeViReal64

C Function Prototype

```
ViStatus niScope_GetAttributeViReal64 (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViReal64* value);
```

Purpose

Queries the value of a ViReal64 attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.

Output

Name	Type	Description
value	ViReal64*	Returns the current value of the attribute; pass the address of a ViReal64 variable.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetAttributeViString

C Function Prototype

```
ViStatus niScope_GetAttributeViString (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViInt32 bufSize, ViChar value[]);
```

Purpose

Queries the value of a ViString attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

You must provide a ViChar array to serve as a buffer for the value. You pass the number of bytes in the buffer as the bufSize. If the current value of the attribute, including the terminating NUL byte, is larger than the size you indicate in the bufSize, the function copies (bufSize - 1) bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the bufSize you must pass to get the entire value. For example, if the value is 123456 and the bufSize is 4, the function places

123 into the buffer and returns 7. If you want to call this function just to get the required buffer size, you can pass 0 for the `bufSize` and `VI_NULL` for the `value`.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>channelList</code>	<code>ViConstString</code>	The channel to configure. For more information, refer to Channel String Syntax .
<code>attributeID</code>	<code>ViAttr</code>	The ID of an attribute.
<code>bufSize</code>	<code>ViInt32</code>	The number of bytes in the <code>ViChar</code> array you specify for <code>value</code> .
Output		
Name	Type	Description
<code>value[]</code>	<code>ViChar</code>	The buffer in which the function returns the current value of the attribute; the buffer must be of type <code>ViChar</code> and have at least as many bytes as indicated in the <code>bufSize</code> .

Return Value

Name	Type	Description								
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetAttributeViBoolean

C Function Prototype

```
ViStatus niScope_GetAttributeViBoolean (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViBoolean* value);
```

Purpose

Queries the value of a ViBoolean attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
Output		
Name	Type	Description
value	ViBoolean*	Returns the current value of the attribute; pass the address of a ViBoolean variable.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetAttributeViSession

C Function Prototype

```
ViStatus niScope_GetAttributeViSession (ViSession vi, ViConstString channelList,
ViAttr attributeID, ViSession* value);
```

Purpose

Queries the value of a ViSession attribute. You can use this function to get the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.

Output

Name	Type	Description
value	ViSession*	Returns the current value of the attribute; pass the address of a ViSession variable.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CheckAttributeViInt32

C Function Prototype

```
ViStatus niScope_CheckAttributeViInt32 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViInt32 value);
```

Purpose

Verifies the validity of a value you specify for a ViInt32 attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViInt32	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CheckAttributeViInt64

C Function Prototype

```
ViStatus niScope_CheckAttributeViInt64 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViInt64 value);
```

Purpose

Verifies the validity of a value you specify for a ViInt64 attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViInt64	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CheckAttributeViReal64

C Function Prototype

```
ViStatus niScope_CheckAttributeViReal64 (ViSession vi, ViConstString channelList, ViAttr attributeID, ViReal64 value);
```

Purpose

Verifies the validity of a value you specify for a ViReal64 attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViReal64	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CheckAttributeViString

C Function Prototype

```
ViStatus niScope_CheckAttributeViString (ViSession vi, ViConstString channelList, ViAttr attributeID, ViConstString value);
```

Purpose

Verifies the validity of a value you specify for a ViString attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViConstString	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CheckAttributeViBoolean

C Function Prototype

```
ViStatus niScope_CheckAttributeViBoolean (ViSession vi, ViConstString channelList, ViAttr attributeID, ViBoolean value);
```

Purpose

Verifies the validity of a value you specify for a ViBoolean attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute
value	ViBoolean	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CheckAttributeViSession

C Function Prototype

```
ViStatus niScope_CheckAttributeViSession (ViSession vi, ViConstString channelList, ViAttr attributeID, ViSession value);
```

Purpose

Verifies the validity of a value you specify for a ViSession attribute.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
attributeID	ViAttr	The ID of an attribute.
value	ViSession	The value that you want to verify for the attribute. Some values might not be valid depending on the current settings of the instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_Abort

C Function Prototype

```
ViStatus niScope_Abort (ViSession vi);
```

Purpose

Aborts an acquisition and returns the digitizer to the Idle state. Call this function if the digitizer times out waiting for a trigger.

Related topics:

- [NI-SCOPE Programming Flow](#)
- [Acquisition Functions](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
------	------	-------------

Status ViStatus Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_AcquisitionStatus

C Function Prototype

```
ViStatus niScope_AcquisitionStatus (ViSession vi, ViInt32* status);
```

Purpose

Returns status information about the acquisition to the status output parameter.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Output

Name	Type	Description
status	ViInt32*	Returns whether the acquisition is complete, in progress, or unknown.

Defined Values

NISCOPE_VAL_ACQ_COMPLETE

NISCOPE_VAL_ACQ_IN_PROGRESS

NISCOPE_VAL_ACQ_STATUS_UNKNOWN

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_Commit

C Function Prototype

```
ViStatus niScope_Commit (ViSession vi);
```

Purpose

Commits to hardware all the parameter settings associated with the task. Use this function if you want a parameter change to be immediately reflected in the hardware.

Related topics:

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [NI-SCOPE Programming Flow](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_Fetch

C Function Prototype

```
ViStatus niScope_Fetch (ViSession vi, ViConstString channelList, ViReal64 timeout,
ViInt32 numSamples, ViReal64* wfm, struct niScope_wfmInfo* wfmInfo);
```

Purpose

Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This function returns scaled voltage waveforms.

This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.



Notes You can use [niScope_Read](#) instead of this function. [niScope_Read](#) starts an acquisition on all enabled channels, waits for the acquisition to complete, and returns the waveform for the specified channel. Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Fetching Data](#)
- [Acquiring Data Continuously](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.
Output		
Name	Type	Description
wfm	ViReal64*	<p>Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms.</p> <p>NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:</p> <p>index 0 = record 0, channel 0</p> <p>index x = record 0, channel 1</p> <p>index 2x = record 1, channel 0</p> <p>index 3x = record 1, channel 1</p> <p>Where x = the record length</p>
wfmInfo	struct niScope_wfmInfo*	<p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between

records and acquisitions; devices that do not support this parameter use 0 for this output.

- `xIncrement`—the time between points in the acquired waveform in seconds
- `actualSamples`—the actual number of samples fetched and placed in the waveform array
- `gain`—the gain factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

- `offset`—the offset factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

Call `niScope_ActualNumWfms` to determine the size of this array.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_FetchComplex

C Function Prototype

```
niScope_FetchComplex ( ViSession vi, ViConstString channelList, ViReal64 timeout,
ViInt32 numSamples, NIComplexNumber* wfm, struct niScope_wfmInfo* wfmInfo);
```

Purpose

Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.
Output		
Name	Type	Description
wfm	NIComplexNumber*	Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms.
wfmInfo	struct niScope_wfmInfo*	Returns an array of structures with the following timing and scaling information about each waveform: <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

- `xIncrement`—the time between points in the acquired waveform in seconds
- `actualSamples`—the actual number of samples fetched and placed in the waveform array
- `gain`—the gain factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

- `offset`—the offset factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

Call [niScope_ActualNumWfms](#) to determine the size of this array.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_FetchComplexBinary16

C Function Prototype

```
niScope_FetchComplexBinary16 ( ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 numSamples, NIComplexI16* wfm, Struct niScope_wfmlInfo* wfmlInfo);
```


Purpose

Retrieves data from single channels and records. Returns a one-dimensional array of complex binary 16-bit waveforms.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.
Output		
Name	Type	Description
wfm	NIComplexI16*	Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms.
wfmInfo	struct niScope_wfmInfo*	Returns an array of structures with the following timing and scaling information about each waveform: <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output. ▪ xIncrement—the time between points in the acquired waveform in seconds

- `actualSamples`—the actual number of samples fetched and placed in the waveform array
- `gain`—the gain factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

- `offset`—the offset factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

Call [`niScope_ActualNumWfms`](#) to determine the size of this array.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

`niScope_FetchBinary8`

C Function Prototype

```
ViStatus niScope_FetchBinary8 (ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 numSamples, ViInt8* wfm, struct niScope_wfmlInfo* wfmlInfo);
```

Purpose

Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Refer to [Using Fetch Functions](#) for more information on using this function.



Note Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.

Output

Name	Type	Description
wfm	ViInt8*	Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values: index 0 = record 0, channel 0 index x = record 0, channel 1

wfmlInfo	struct niScope_ wfmlInfo*	<p>index 2x = record 1, channel 0</p> <p>index 3x = record 1, channel 1</p> <p>Where x = the record length</p> <p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output. ▪ xIncrement—the time between points in the acquired waveform in seconds ▪ actualSamples—the actual number of samples fetched and placed in the waveform array ▪ gain—the gain factor of the given channel; useful for scaling binary data with the following formula: <p style="margin-left: 40px;">voltage = binary data × gain factor + offset</p> <ul style="list-style-type: none"> ▪ offset—the offset factor of the given channel; useful for scaling binary data with the following formula: <p style="margin-left: 40px;">voltage = binary data × gain factor + offset</p> <p>Call niScope_ActualNumWfms to determine the size of this array.</p>
----------	---------------------------------	---

Return Value

Name	Type	Description				
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;">Value</th> <th style="width: 50%; text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td style="height: 20px;"> </td> <td> </td> </tr> </tbody> </table>	Value	Meaning		
Value	Meaning					

0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_FetchBinary16

C Function Prototype

```
ViStatus niScope_FetchBinary16 (ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 numSamples, ViInt16* wfm, struct niScope_wfmInfo* wfmInfo);
```

Purpose

Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Refer to [Using Fetch Functions](#) for more information on using this function.



Note Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.

Output

Name	Type	Description
wfm	VlInt16*	<p>Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms.</p> <p>NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:</p> <p>index 0 = record 0, channel 0</p> <p>index x = record 0, channel 1</p> <p>index 2x = record 1, channel 0</p> <p>index 3x = record 1, channel 1</p> <p>Where x = the record length</p>
wfmInfo	struct niScope_ wfmInfo*	<p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output. ▪ xIncrement—the time between points in the acquired waveform in seconds ▪ actualSamples—the actual number of samples fetched and placed in the waveform array ▪ gain—the gain factor of the given channel; useful for scaling binary data with the following formula: $\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$ <ul style="list-style-type: none"> ▪ offset—the offset factor of the given channel; useful for scaling binary data with the following formula: $\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$

Call [niScope_ActualNumWfms](#) to determine the size of this array.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_FetchBinary32

C Function Prototype

```
ViStatus niScope_FetchBinary32 (ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 numSamples, ViInt32* wfm, struct niScope_wfmlInfo* wfmlInfo);
```

Purpose

Retrieves data from a previously initiated acquisition and returns binary 32-bit waveforms. This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Refer to [Using Fetch Functions](#) for more information on using this function.



Note Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
------	------	-------------

vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
numSamples	ViInt32	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.

Output

Name	Type	Description
wfm	ViInt32*	<p>Returns an array whose length is the numSamples times number of waveforms. Call <code>niScope_ActualNumWfms</code> to determine the number of waveforms.</p> <p>NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:</p> <p>index 0 = record 0, channel 0</p> <p>index x = record 0, channel 1</p> <p>index 2x = record 1, channel 0</p> <p>index 3x = record 1, channel 1</p> <p>Where x = the record length</p>
wfmInfo	struct niScope_ wfmInfo*	<p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output.

- xIncrement—the time between points in the acquired waveform in seconds
- actualSamples—the actual number of samples fetched and placed in the waveform array
- gain—the gain factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

- offset—the offset factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

Call [niScope_ActualNumWfms](#) to determine the size of this array.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetNormalizationCoefficients

```
ViStatus _VI_FUNC niScope_GetNormalizationCoefficients (ViSession vi,
ViConstString channelList, ViInt32 bufferSize, struct niScope_coefficientInfo
coefficientInfo[], ViInt32* numberOfCoefficientSets);
```

Purpose

Returns coefficients that can be used to convert binary data to normalized and calibrated data.

Refer to [Scaling and Normalization of Binary Data](#) for more information about how to use this function.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScopeInit that identifies a particular instrument session.
channel List	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
bufferSize	ViInt32	The array size for the <code>coefficientInfo</code> parameter. To determine the size of the buffer to allocate for <code>coefficientInfo</code> , pass a value of 0 to the <code>bufferSize</code> parameter and a value of NULL to the <code>coefficientInfo</code> parameter. In this case, the return value of the <code>numberOfCoefficientSets</code> parameter is the size of the array necessary to hold the coefficient structures. Allocate an array of <code>niScope_coefficientInfo</code> structures of this size, then call this function again (with the correct <code>bufferSize</code> parameter) to retrieve the actual values.
coefficientInfo	struct niScope_coefficientInfo	An array of structures containing gain and offset coefficients for a given channel.
Output		
Name	Type	Description
numberOfCoefficientSets	ViInt32*	Returns the number of coefficient sets returned in the <code>coefficientInfo</code> array.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetScalingCoefficients

```
ViStatus _VI_FUNC niScope_GetScalingCoefficients (ViSession vi, ViConstString
channelList, ViInt32 bufferSize, struct niScope_coefficientInfo coefficientInfo[],
ViInt32* numberOfCoefficientSets);
```

Purpose

Returns coefficients that can be used to scale binary data to volts.

Refer to [Scaling and Normalization of Binary Data](#) for more information about how to use this function.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
bufferSize	ViInt32	The array size for the <code>coefficientInfo</code> parameter.

To determine the size of the buffer to allocate for `coefficientInfo`, pass a value of 0 to the `bufferSize` parameter and a value of NULL to the `coefficientInfo` parameter. In this case, the return value of the `numberOfCoefficientSets` parameter is the size of the array necessary to hold the coefficient structures. Allocate an array of `niScope_coefficientInfo` structures of this size, then call this function again (with the correct `bufferSize` parameter) to retrieve the actual values.

`coefficientInfo` struct
 `niScope_coefficientInfo` An array of structures containing gain and offset coefficients for a given channel.

Output

Name	Type	Description
<code>numberOfCoefficientSets</code>	<code>ViInt32</code>	Returns the number of coefficient sets returned in the <code>coefficientInfo</code> array.

Return Value

Name	Type	Description
Status	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_InitiateAcquisition

C Function Prototype

`ViStatus niScope_InitiateAcquisition (ViSession vi);`

Purpose

Initiates a waveform acquisition.

After calling this function, the digitizer leaves the Idle state and waits for a trigger. The digitizer acquires a waveform for each channel you enable with [niScope_ConfigureVertical](#).

Related topics:

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)
- [NI-SCOPE Programming Flow](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_Read

C Function Prototype

```
ViStatus niScope_Read (ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 numSamples, ViReal64* wfm, struct niScope_wfmInfo* wfmInfo);
```

Purpose

Initiates an acquisition, waits for it to complete, and retrieves the data. The process is similar to calling `niScope_InitiateAcquisition`, `niScope_AcquisitionStatus`, and `niScope_Fetch`. The only difference is that with `niScope_Read`, you enable all channels specified with `channelList` before the acquisition; in the other method, you enable the channels with `niScope_ConfigureVertical`.

This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.



Note Some functionality is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Related topics:

- [Acquisition Functions](#)
- [NI-SCOPE Programming Flow](#)

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>channelList</code>	<code>ViConstString</code>	The channel to configure. For more information, refer to Channel String Syntax .
<code>timeout</code>	<code>ViReal64</code>	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
<code>numSamples</code>	<code>ViInt32</code>	The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. If it fails to complete within the timeout period, the function returns an error.

Output

Name	Type	Description
------	------	-------------

wfm	ViReal64*	<p>Returns an array whose length is the numSamples times number of waveforms. Call niScope_ActualNumWfms to determine the number of waveforms.</p> <p>NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:</p> <p>index 0 = record 0, channel 0</p> <p>index x = record 0, channel 1</p> <p>index $2x$ = record 1, channel 0</p> <p>index $3x$ = record 1, channel 1</p> <p>Where x = the record length</p>
wfmInfo	struct niScope_ wfmInfo*	<p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> ▪ relativeInitialX—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ absoluteInitialX—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output. ▪ xIncrement—the time between points in the acquired waveform in seconds ▪ actualSamples—the actual number of samples fetched and placed in the waveform array ▪ gain—the gain factor of the given channel; useful for scaling binary data with the following formula: $\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$ <ul style="list-style-type: none"> ▪ offset—the offset factor of the given channel; useful for scaling binary data with the following formula: $\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$ <p>Call niScope_ActualNumWfms to determine the size of this array.</p>

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_AddWaveformProcessing

C Function Prototype

```
ViStatus niScope_AddWaveformProcessing (ViSession vi, ViConstString channelList, ViInt32 measFunction);
```

Purpose

Adds one measurement to the list of processing steps that are completed before the measurement. The processing is added on a per channel basis, and the processing measurements are completed in the same order they are registered. All measurement library parameters—the attributes starting with `NISCOPE_ATTR_MEAS`—are cached at the time of registering the processing, and this set of parameters is used during the processing step. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing is done before any other measurements.

Related topics:

- [Array Measurements in NI-SCOPE](#)
- [Scalar Measurements in NI-SCOPE](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
measFunction	ViInt32	The array measurement to add.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ClearWaveformMeasurementStats

C Function Prototype

```
ViStatus niScope_ClearWaveformMeasurementStats (ViSession vi, ViConstString channelList, ViInt32 clearableMeasurementFunction);
```

Purpose

Clears the waveform stats on the channel and measurement you specify. If you want to clear all of the measurements, use `NISCOPE_VAL_ALL_MEASUREMENTS` in the `clearableMeasurementFunction` parameter.

Every time a measurement is called, the statistics information is updated, including the min, max, mean, standard deviation, and number of updates. This information

is fetched with [niScope_FetchMeasurementStats](#). The multi-acquisition array measurements are also cleared with this function.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
clearableMeasurementFunction	ViInt32	The scalar measurement or array measurement to clear the stats for.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ClearWaveformProcessing

C Function Prototype

```
ViStatus niScope_ClearWaveformProcessing (ViSession vi, ViConstString channelList);
```

Purpose

Clears the list of processing steps assigned to the given channel. The processing is added using the [niScope_AddWaveformProcessing](#) function, where the processing steps are completed in the same order in which they are registered. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing is also done before any other measurements.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_FetchArrayMeasurement

C Function Prototype

```
ViStatus niScope_FetchArrayMeasurement (ViSession vi, ViConstString channelList,
ViReal64 timeout, ViInt32 arrayMeasFunction, ViInt32 measWfmSize, ViReal64*
measWfm, struct niScope_wfmlInfo* wfmlInfo);
```

Purpose

Obtains a waveform from the digitizer and returns the specified measurement array. This function may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.



Note Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
arrayMeasFunction	ViInt32	The array measurement to perform.
measWfmSize	ViInt32	The maximum number of samples returned in the measurement waveform array for each waveform measurement. Use niScope_ActualMeasWfmSize to determine the number of available samples.



Note Use the attribute [NISCOPE_ATTR_FETCH_MEAS_NUM_SAMPLES](#) to set the number of samples to fetch when performing a

measurement. For more information about when to use this attribute, refer to the NI KnowledgeBase.

Output

Name	Type	Description
measWfm	ViReal64*	<p>Returns an array whose length is the number of waveforms times <code>measWfmSize</code>; call <code>niScope_ActualNumWfms</code> to determine the number of waveforms; call <code>niScope_ActualMeasWfmSize</code> to determine the size of each waveform. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with channel list of 0, 1, you would have the following index values:</p> <p>index 0 = record 0, channel 0</p> <p>index x = record 0, channel 1</p> <p>index 2x = record 1, channel 0</p> <p>index 3x = record 1, channel 1</p> <p>Where x = the record length</p>
wfmInfo	struct niScope_wfmInfo*	<p>Returns an array of structures with the following timing and scaling information about each waveform:</p> <ul style="list-style-type: none"> ▪ <code>relativeInitialX</code>—the time (in seconds) from the trigger to the first sample in the fetched waveform ▪ <code>absoluteInitialX</code>—timestamp (in seconds) of the first fetched sample. This timestamp is comparable between records and acquisitions; devices that do not support this parameter use 0 for this output. ▪ <code>xIncrement</code>—the time between points in the acquired waveform in seconds ▪ <code>actualSamples</code>—the actual number of samples fetched and placed in the waveform array ▪ <code>gain</code>—the gain factor of the given channel; useful for scaling binary data with the following formula: <p>$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$</p>

- **offset**—the offset factor of the given channel; useful for scaling binary data with the following formula:

$$\text{voltage} = \text{binary data} \times \text{gain factor} + \text{offset}$$

Call [niScope_ActualNumWfms](#) to determine the size of this array.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_FetchMeasurement

C Function Prototype

```
ViStatus niScope_FetchMeasurement (ViSession vi, ViConstString channelList, ViReal64 timeout, ViInt32 scalarMeasFunction, ViReal64* result);
```

Purpose

Fetches a waveform from the digitizer and performs the specified waveform measurement. Refer to [Using Fetch Functions](#) for more information.

Many of the measurements use the low, mid, and high reference levels. You configure the low, mid, and high references by using [NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL](#), [NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL](#), and [NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL](#) to set each channel differently.

Related topics:

- [Making Waveform Measurements](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
scalarMeasFunction	ViInt32	The scalar measurement to be performed.
Output		
Name	Type	Description
result	ViReal64*	Contains an array of all measurements acquired; call niScope_ActualNumWfms to determine the array length.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_FetchMeasurementStats

C Function Prototype

```
ViStatus niScope_FetchMeasurementStats (ViSession vi, ViConstString channelList,
ViReal64 timeout, ViInt32 scalarMeasFunction, ViReal64* result, ViReal64* mean,
ViReal64* stdev, ViReal64* min, ViReal64* max, ViInt32*numInStats);
```

Purpose

Obtains a waveform measurement and returns the measurement value. This function may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.

You specify a particular measurement type, such as rise time, frequency, or voltage peak-to-peak. The waveform on which the digitizer calculates the waveform measurement is from an acquisition that you previously initiated. The statistics for the specified measurement function are returned, where the statistics are updated once every acquisition when the specified measurement is fetched by any of the Fetch Measurement functions. If a Fetch Measurement function has not been called, this function fetches the data on which to perform the measurement. The statistics are cleared by calling [niScope_ClearWaveformMeasurementStats](#). Refer to [Using Fetch Functions](#) for more information on incorporating fetch functions in your application.

Many of the measurements use the low, mid, and high reference levels. You configure the low, mid, and high references with [NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL](#), [NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL](#), and [NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL](#) to set each channel differently.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
scalarMeasFunction	ViInt32	The scalar measurement to be performed on each fetched waveform.

Output

Name	Type	Description
result	ViReal64*	Returns the resulting measurement
mean	ViReal64*	Returns the mean scalar value, which is obtained by averaging each <code>niScope_FetchMeasurementStats</code> call.
stdev	ViReal64*	Returns the standard deviation of the most recent <code>numInStats</code> measurements.
min	ViReal64*	Returns the smallest scalar value acquired (the minimum of the <code>numInStats</code> measurements).
max	ViReal64*	Returns the largest scalar value acquired (the maximum of the <code>numInStats</code> measurements).
numInStats	ViInt32*	Returns the number of times <code>niScope_FetchMeasurementStats</code> has been called.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ReadMeasurement

C Function Prototype

```
ViStatus_VI_FUNC niScope_ReadMeasurement (ViSession vi, ViConstString
channelList, ViReal64 timeout, ViInt32 scalarMeasFunction, ViReal64* result);
```

Purpose

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records.

Refer to [Using Fetch Functions](#) for more information.

Many of the measurements use the low, mid, and high reference levels. You configure the low, mid, and high references by using [NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL](#), [NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL](#), and [NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL](#) to set each channel differently.

Related topics:

- [Making Waveform Measurements](#)
- [Acquisition Functions](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
timeout	ViReal64	The time to wait in seconds for data to be acquired; using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.
scalarMeasFunction	ViInt32	The scalar measurement to be performed
Output		

Name	Type	Description
result	ViReal64*	Contains an array of all measurements acquired. Call niScope_ActualNumWfms to determine the array length.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalSelfCalibrate

C Function Prototype

```
ViStatus niScope_CalSelfCalibrate (ViSession sessionHandle, ViConstString channelList, ViInt32 option);
```

Purpose

Self-calibrates most NI digitizers, including all SMC-based devices. To verify that your digitizer supports self-calibration, refer to [Features Supported by Device](#).

For SMC-based digitizers, if the self-calibration is performed successfully in a regular session, the calibration constants are immediately stored in the self-calibration area of the EEPROM. If the self-calibration is performed in an external calibration session, the calibration constants take effect immediately for the duration of the session.

However, they are not stored in the EEPROM until [niScope_CalEnd](#) is called with action set to `NISCOPE_VAL_ACTION_STORE` and no errors occur.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channelList	ViConstString	The channel to configure. For more information, refer to Channel String Syntax .
option	ViInt32	The calibration option. Use <code>VI_NULL</code> for a normal self-calibration operation or <code>NISCOPE_VAL_CAL_RESTORE_EXTERNAL_CALIBRATION</code> to restore the previous calibration.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustAccessoryGainAndOffset

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustAccessoryGainAndOffset ( ViSession
sessionHandle, ViConstString channelName, ViReal64 positiveFS, ViReal64 ground,
ViReal64 negativeFS );
```

Purpose

For the NI 5900, this function externally calibrates the gain and offset.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle returned by niScope_CalStart .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax .
positiveFS	ViReal64	This value is calculated during the external calibration procedure.
ground	ViReal64	This value is calculated during the external calibration procedure.
negativeFS	ViReal64	This value is calculated during the external calibration procedure.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustCompensationAttenuator

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustCompensationAttenuator (ViSession sessionHandle, ViConstString channelName, ViReal64 range);
```

Purpose

For the NI 5132/5133, this function externally calibrates the compensation attenuator. Refer to the **NI 5132/5133 Calibration Procedure** for more information.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax .
range	ViReal64	The voltage range to calibrate.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustDCM

C Function Prototype

```
ViStatus niScope_CalAdjustDCM (ViSession sessionHandle, ViConstString
channelName, ViReal64 stimulusFreq);
```

Purpose

For the NI 5104/5105 digitizer, calibrates the external clock digital clock managers (DCMs). DCM calibration ensures that data can be sampled at the correct time in the clock period.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart .
channelName	ViConstString	This parameter is ignored.
stimulusFreq	ViReal64	The external stimulus applied to the digitizer. Default value: 0

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustFrequencyResponse

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustFrequencyResponse ( ViSession
sessionHandle, ViConstString channelName, ViReal64 range, ViReal64 stimulusFreq,
ViReal64 stimulusAmp );
```

Purpose

For the NI 5622, calibrates the frequency response of the device. The function is called multiple times, one for each frequency point in the sweep. The user must supply the stimulus frequency (in hertz) and stimulus amplitude (in volts) of the input signal.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle returned by niScope_CalStart .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax .
range	ViReal64	The voltage range to calibrate.
stimulusFreq	ViReal64	The frequency of the external stimulus applied to the digitizer.
stimulus	ViReal64	The peak voltage of the signal applied to the digitizer.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>			Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CalAdjustOffsetRange

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustOffsetRange ( ViSession sessionHandle,
ViConstString channelName, ViReal64 range, ViReal64 stimulus );
```

Purpose

For the NI 5132/5133, this function externally calibrates the vertical offset. Refer to the **NI 5132/5133 Calibration Procedure** for the appropriate stimulus voltages.

Parameters

Input		
Name	Type	Description

sessionHandle	ViSession	The session handle you obtain from niScope_CalStart .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax .
range	ViReal64	The voltage range to calibrate.
stimulus	ViReal64	The peak voltage of the applied signal.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:


Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustRange

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustRange (ViSession sessionHandle,
ViConstString channelName, ViReal64 range, ViReal64 stimulus);
```

Purpose

 **Note** Use this function only when following the device calibration procedure.

Before calling this function, connect an accurate DC reference voltage to one of the digitizer's channels. Use the function parameters to indicate the voltage and channel.

- **SMC-Based Digitizers**

For SMC-based digitizers, this function calibrates the range (that is, gain) accuracy of the entire device using the specified channel. The function must be called twice in the same session with two different stimulus values. The

stimulus values should be several volts apart, but within ± 4 V; the recommended values are +4.0 V and 0.0 V. The two calls should use the same channel—it is not necessary to repeat the procedure on the remaining channel. This function adjusts the gain and range calibration constants so that the digitizer reads exactly the correct difference between the two stimulus voltages. This calibrates the onboard voltage reference against the external voltage reference. Calibrating with a voltage difference instead of a single reading eliminates any error due to offset.

The new calibration constants for the digitizer take effect immediately for the duration of the external calibration session. The constants are written to the EEPROM if you call `niScope_CalEnd` with no errors and with action set to `NI_SCOPE_VAL_ACTION_STORE`.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from <code>niScope_CalStart</code> .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax
range	ViReal64	The voltage range to calibrate.
stimulus	ViReal64	The peak voltage of the applied signal.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalAdjustVCXO

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalAdjustVCXO (ViSession sessionHandle, ViReal64 stimulusFreq);
```

Purpose



Note Use this function only when following the device calibration procedure.

For SMC-based digitizers, this function calibrates the sample rate of the digitizer. The function adjusts the frequency of the oscillator (VCXO) that serves as the digitizer's onboard sample rate timebase. Check the calibration procedure specific to your device and set the value of the `stimulusFreq` parameter. Before calling this function, connect an accurate, stable reference signal to channel 0. (The channel used is not configurable.) The function adjusts frequency calibration constants until the digitizer exactly measures the frequency of the reference signal. The new calibration constants take effect immediately for the duration of the external calibration session. The constants are written to the EEPROM if you call [niScope_CalEnd](#) with no errors and with action set to `NISCOPE_VAL_CAL_ACTION_STORE`.

Parameters

Input

Name	Type	Description
<code>sessionHandle</code>	<code>ViSession</code>	The session handle you obtain from niScope_CalStart .
<code>stimulusFreq</code>	<code>ViReal64</code>	The frequency of the reference clock present on channel 0.

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalChangePassword

C Function Prototype

```
ViStatus_VI_FUNC niScope_CalChangePassword (ViSession sessionHandle,
ViConstString oldPassword, ViConstString newPassword);
```

Purpose

Verifies the old password against the one stored in the EEPROM. If the two match, the function stores the new password in the EEPROM. The password is stored as four characters, but shorter strings are acceptable. For most digitizers, the default password is the empty string (" "). For SMC-based digitizers, the default password is "NI". If you forget your password, call National Instruments for assistance.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart .
oldPassword	ViConstString	The previous password is verified against what is stored in the EEPROM.
newPassword	ViConstString	The new password that is written to the EEPROM.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError .

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalEnd

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalEnd ( ViSession sessionHandle, ViInt32 action );
```

Purpose

Closes an external calibration session. If `action` is set to `NISCOPE_VAL_ACTION_ABORT`, the session is closed and the new calibration constants are lost. Some devices may write to the EEPROM during calibration, in which case the Abort Calibration action restores the EEPROM to its original state. It is, therefore, very important to call `niScope_CalEnd` each time `niScope_CalStart` is called, even if an error occurs during calibration.

If `action` is set to `NISCOPE_VAL_ACTION_STORE`, the new calibration constants are stored in the EEPROM. For most digitizers, the current system date and the incremented calibration count are also stored; for SMC-based digitizers, the current system date and onboard temperature are stored.

Parameters

Input		
Name	Type	Description
<code>sessionHandle</code>	<code>ViSession</code>	The session handle returned by <code>niScope_CalStart</code> .
<code>action</code>	<code>ViInt32</code>	<code>NISCOPE_VAL_CAL_ACTION_STORE</code> or <code>NISCOPE_VAL_CAL_ACTION_ABORT</code>

Return Value

Name	Type	Description
------	------	-------------

Status ViStatus Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalFetchCount

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchCount ( ViSession sessionHandle, ViInt32 whichOne, ViInt32* calibrationCount );
```

Purpose

Returns the calibration count, which is the number of times the device has been calibrated.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart or niScope_Init .
whichOne	ViInt32	Which type of calibration count to store: self-calibration or external calibration count.

Output

Name	Type	Description
calibrationCount	ViInt32*	The number of calibrations performed on this device.

Return Value

Name	Type	Description
------	------	-------------

Status `ViStatus` Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalFetchDate

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchDate (ViSession sessionHandle, ViInt32 whichOne, ViInt32* year, ViInt32* month, ViInt32* day);
```

Purpose

Returns the last self-calibration, external calibration, or manufacture date.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart or niScope_init .
whichOne	ViInt32	The type of calibration for which the information is stored: self-calibration, external calibration, or manufacture date.

Output

Name	Type	Description
year	ViInt32*	The year of the last calibration, such as 2003.
month	ViInt32*	The month of the last calibration (1–12).
day	ViInt32*	The day of the last calibration (1–31).

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CalFetchMiscInfo

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchMiscInfo ( ViSession sessionHandle, ViChar*
miscInfo );
```

Purpose

Returns the miscellaneous information stored in the EEPROM using [niScope_CalStoreMiscInfo](#).

Parameters

Input		
Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart or niScope_e_init .
Output		
Name	Type	Description
miscInfo	ViChar*	4 characters stored in the EEPROM.

Return Value

Name	Type	Description
------	------	-------------

Status ViStatus Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalFetchTemperature

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalFetchTemperature (ViSession sessionHandle, ViInt32 whichOne, ViReal64* temperature);
```

Purpose

For SMC-based digitizers, returns the onboard temperature of the digitizer at the time of the last self-calibration or external calibration, in degrees Celsius. The temperature returned by this function is an onboard temperature read from a sensor on the surface of the digitizer. This temperature should not be confused with the environmental temperature of the digitizer's surroundings. During operation, the onboard temperature is normally higher than the environmental temperature.

Temperature-sensitive parameters are calibrated during self-calibration. Therefore, the self-calibration temperature is usually the more important one to read.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart .
whichOne	ViInt32	Defined Values NISCOPE_VAL_CAL_SELF NISCOPE_VAL_CAL_EXTERNAL

Output

Name	Type	Description
temperature	ViReal64*	The returned temperature of last calibration, in °C.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalMeasureRISDistribution

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalMeasureRISDistribution (ViSession sessionHandle,
ViConstString channelName, ViInt32 maxTime, ViReal64* minBinPercent, ViInt32
distributionSize, ViInt32* distribution);
```

Purpose

Calls [niScope_Read](#) 2,000 times to take an acquisition from the specified channel and retrieve the initial X value, which includes the time-to-digital (TDC) conversion. The TDC should be a uniform distribution between two sample points because triggers should occur randomly. To test this distribution, the distribution of initial X values is created. The percentage of triggers in the smallest bin of this distribution is returned for comparison to a specification to determine if RIS is operating correctly. The distribution parameter must be declared as an array of distributionSize to return the distribution. Optionally, setting distribution to VI_NULL specifies that the distribution is not returned.

Related topics:

- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

Parameters

Input		
Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart or niScope_init .
channelName	ViConstString	The name of the channel to calibrate. For more information, refer to Channel String Syntax .
maxTime	ViInt32	The maximum time in ms for each acquisition.
minBinPercent	ViReal64*	The percentage (0–1) of triggers in the least full bin.
Output		
Name	Type	Description
distributionSize	ViInt32	The number of bins for distribution.
distribution	ViInt32*	The array for distribution; use <code>VI_NULL</code> for do not return.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_CalSetAccessorySource

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalSetAccessorySource ( ViSession sessionHandle,
ViConstString channelName, ViInt32 calSource );
```

Purpose

For the NI 5900, this function sets the calibration source for the device.

Parameters

Input		
Name	Type	Description
sessionHandle	ViSession	The session handle returned by niScope_CalStart .
channelName	ViConstString	The string name of the channel to calibrate. For more information, refer to Channel String Syntax .
calSource	ViInt32	The calibration signal source.

Defined Values

NISCOPE_VAL_CAL_SOURCE_GROUND

NISCOPE_VAL_CAL_SOURCE_POSITIVEFS

NISCOPE_VAL_CAL_SOURCE_NEGATIVEFS

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values

Errors

niScope_CalStart

C Function Prototype

```
ViStatus_VI_FUNC niScope_CalStart(ViRsrc resourceName, ViConstString password, ViSession* sessionHandle);
```

Purpose

Opens an external calibration session and produces a calibration session handle that is required by the external calibration functions. All other calibration functions, such as verification and fetch VIs, work with both a calibration session and a session handle obtained from [niScope_init](#). Acceptable session handles are documented for each function in the calibration procedure documents.

By default, the calibration password for the SMC-based digitizers is "NI". The password is stored in the EEPROM as an array of four characters. Nonprintable characters are allowed, but the array is padded with NULLs after the first NULL is found. This padding allows strings of less than four characters to be valid passwords. The password is verified against the password stored in the EEPROM. You can change the password from the default by calling [niScope_CalChangePassword](#).

Only one session handle can be obtained at a time, and every session started with [niScope_CalStart](#) must be closed by calling [niScope_CalEnd](#). If you fail to close the session, you must unload the `niScope_32.dll` by closing your application or application development environment (ADE) before you can open another session.

If an error occurs during calibration, call [niScope_errorHandler](#) to get the error message text. You can call [niScope_CalEnd](#) with `action` set to `NISCOPE_VI_CAL_ACTION_ABORT` to end the session without updating the EEPROM. For SMC-based digitizers, the EEPROM is also not updated if an error occurs.

Parameters

Input

Name	Type	Description
------	------	-------------

resourceName	ViRsrc	The device number assigned by Measurement & Automation Explorer (MAX).
password	ViConstString	The password you use, which is verified against the password stored in the EEPROM.

Output

Name	Type	Description
sessionHandle	ViSession*	The returned session handle.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CalStoreMiscInfo

C Function Prototype

```
ViStatus _VI_FUNC niScope_CalStoreMiscInfo (ViSession sessionHandle,
ViConstString miscInfo);
```

Purpose

Stores miscellaneous information in the EEPROM. For example, you can store an operator ID for the person or company performing a calibration.

Four characters are stored in the EEPROM, and nonprintable characters are valid. However, NULL is treated as an end-of-string marker, and all characters following the first NULL are also set to NULL.

Parameters

Input

Name	Type	Description
sessionHandle	ViSession	The session handle you obtain from niScope_CalStart or niScope_init .
miscInfo	ViConstString	Pointer to 4 characters stored in the EEPROM; can be less than four if NULL terminated.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ResetDevice

C Function Prototype

```
ViStatus niScope_ResetDevice (ViSession vi);
```

Purpose

Performs a hard reset of the device. Acquisition stops, all routes are released, RTSI and PFI lines are tristated, hardware is configured to its default state, and all session attributes are reset to their default state.

Related topics:

- [Thermal Shutdown](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_Disable

C Function Prototype

```
ViStatus niScope_Disable (ViSession vi);
```

Purpose

Aborts any current operation, opens data channel relays, and releases RTSI and PFI lines.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ProbeCompensationSignalStart

C Function Prototype

```
ViStatus niScope_ProbeCompensationSignalStart (ViSession vi);
```


Purpose

Generates a 1 kHz square wave signal for probe compensation.

Most oscilloscopes output the probe compensation signal on PFI 1.

The following oscilloscopes output the probe compensation signal in unique locations.

Device	Output Location	Notes
PXIe-5110 PXIe-5111 PXIe-5113	Probe compensation terminal	The signal at this terminal is enabled by default.
PXIe-5163 PXIe-5164	SMB PFI 0	Though the PFI 0 line is also available from the AUX 0 MHDMR connector of these oscilloscopes, the probe compensation signal is available only from SMB PFI 0.

 **Note** Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors


niScope_ProbeCompensationSignalStop

C Function Prototype

```
ViStatus niScope_ProbeCompensationSignalStop (ViSession vi);
```

Purpose

Disables the 1 kHz square wave signal for probe compensation.

 **Note** Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_IsDeviceReady

C Function Prototype

```
ViStatus niScope_IsDeviceReady (ViRsrc resourceName, ViConstString channelList, ViBoolean* deviceReady);
```

Purpose

Call this function to determine whether the device is ready for use or the device is still undergoing initialization.

Parameters

Input		
Name	Type	Description
resourceName	ViRsrc	resourceName specifies the resource name of the device to initialize.

resourceName Examples

Example#	Device Type	Syntax	Variable
1	NI-DAQmx device	myDAQmxDevice	(myDAQmxDevice = device name)
2	NI-DAQmx device	DAQ:: myDAQmxDevice	(myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ:: 2	(2 = device name)
4	IVI logical name or IVI virtual name	myLogicalName	(myLogicalName = name)

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 1. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on the name in MAX and entering a new name.

An alternate syntax for NI-DAQmx devices consists of DAQ::**NI-DAQmx device name**, as shown in Example 2. This naming convention allows for the use of an NI-DAQmx device in an application that was originally designed for a Traditional NI-DAQ device. For example, if the application expects DAQ::1, you can rename the NI-DAQmx device to 1 in MAX and pass in DAQ::1 for the resource name, as shown in Example 3.

If you use the DAQ::**n** syntax and an NI-DAQmx device name already exists with that same name, the NI-DAQmx device is matched first.

You can also pass in the name of an IVI logical name or an IVI virtual name configured with the IVI Configuration utility, as shown in Example 4. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.



Note NI-DAQmx device names are not case-sensitive. However, all IVI names, such as logical names, are case-sensitive. If you use logical names, driver session names, or virtual names in your program, you must make sure that the name you use matches the name in the IVI Configuration

Store file exactly, without any variations in the case of the characters.

channelList ViConstString Use only "" or a null pointer. If you specify a channel, NI-SCOPE will return an error.

Output

Name	Type	Description
deviceReady	ViBoolean*	Returns True if the device is ready to use, or False if the device is still initializing.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_reset

C Function Prototype

```
ViStatus niScope_reset (ViSession vi);
```

Purpose

Stops the acquisition, releases routes, and all session attributes are reset to their default states.

Parameters

Input

Name	Type	Description
------	------	-------------

vi ViSession The instrument handle you obtain from [niScope_init](#) that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ResetWithDefaults

C Function Prototype

```
ViStatus niScope_Reset_With_Defaults (ViSession vi);
```

Purpose

Performs a software reset of the device, returning it to the default state and applying any initial default settings from the IVI Configuration Store.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information

concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_revision_query

C Function Prototype

```
ViStatus niScope_revision_query (ViSession vi, ViChar
driverRev[IVI_MAX_MESSAGE_BUF_SIZE], ViChar
instrRev[IVI_MAX_MESSAGE_BUF_SIZE]);
```

Purpose

Returns the revision numbers of the instrument driver and instrument firmware.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Output

Name	Type	Description
driverRev	ViChar[]	Returns the instrument driver software revision numbers in the form of a string; you must pass a ViChar array at least <code>IVI_MAX_MESSAGE_BUF_SIZE</code> bytes in length.
instrRev	ViChar[]	Returns the instrument firmware revision numbers in the form of a string; you must pass a ViChar array at least <code>IVI_MAX_MESSAGE_BUF_SIZE</code> bytes in length.

Return Value

Name	Type	Description
------	------	-------------

Status ViStatus Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_self_test

C Function Prototype

```
ViStatus niScope_self_test (ViSession vi, ViInt16* selfTestResult, ViChar
selfTestMessage[IVI_MAX_MESSAGE_BUF_SIZE]);
```

Purpose

Runs the instrument self-test routine and returns the test result(s).

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Output

Name	Type	Description
selfTestResult	ViInt16*	This control contains the value returned from the instrument self-test. Self-Test Code Description 0—Self-test passed 1—Self-test failed
selfTestMessage	ViChar[]	Returns the self-test response string from the instrument. Refer to the device-specific help topics for an explanation of the string contents;

you must pass a ViChar array at least `IVI_MAX_MESSAGE_BUF_SIZE` bytes in length.

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_GetStreamEndpointHandle

ViStatus niScope_GetStreamEndpointHandle (ViSession vi, ViConstString Stream_Name, ViUInt32 *Writer_Handle);

Purpose

Returns a writer endpoint that can be used with NI-P2P to configure a peer-to-peer stream with a digitizer endpoint.

Related topics:

- [Peer-to-Peer Streaming](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
Stream_Name	ViConstString	The stream endpoint FIFO to configure. Refer to the device-specific documentation for peer-to-peer streaming in the High-Speed Digitizers Help for more information.

Output

Name	Type	Description
Writer_Handle	ViUInt32	Returns a reference to a peer-to-peer writer FIFO that can be used to create a peer-to-peer streaming session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CableSenseSignalStart

C Function Prototype

```
ViStatus niScope_CableSenseSignalStart (ViSession Instrument_Handle);
```

Purpose

Generates the CableSense signal on all channels of an oscilloscope for which the signal is enabled, as configured by the [NISCOPE_ATTR_CABLE_SENSE_MODE](#) attribute.



Note The input impedance of the channel(s) to convey the CableSense signal must be set to 50 Ω.

You can call this function only during an acquisition. If you call this function while your oscilloscope is not acquiring, NI-SCOPE generates an error.

Supported Devices

- [PXIe-5110](#)

- [PXle-5111](#)
- [PXle-5113](#)
- [PXle-5160](#)
- [PXle-5162](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_CableSenseSignalStop

C Function Prototype

```
ViStatus niScope_CableSenseSignalStop (ViSession Instrument_Handle);
```

Purpose

Disables the CableSense signal on all channels of an oscilloscope for which the signal is enabled.

Supported Devices

- [PXle-5110](#)
- [PXle-5111](#)
- [PXle-5113](#)
- [PXle-5160](#)
- [PXle-5162](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_errorHandler

C Function Prototype

```
ViStatus niScope_errorHandler (ViSession vi, ViInt32 errorCode, ViChar
errorSource[MAX_FUNCTION_NAME_SIZE], ViChar
errorDescription[MAX_ERROR_DESCRIPTION]);
```

Purpose

Takes the error code returned by NI-SCOPE functions and returns the interpretation as a user-readable string.



Note You can pass `VI_NULL` as the instrument handle, which is useful to interpret errors after `niScope_init` has failed.

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>errorCode</code>	<code>ViStatus</code>	The error code that is returned from any of the instrument driver functions.
<code>errorSource[]</code>	<code>ViChar</code>	Specifies the function in which the error occurred. You can pass in a string no longer than <code>MAX_FUNCTION_NAME_SIZE</code> . If you pass in a valid string, this source is included in the <code>errorDescription</code> string. For example: "Error <errorCode> at <errorSource>" If you pass in <code>NULL</code> or an empty string, this parameter is ignored.

Output

Name	Type	Description
<code>errorDescription[]</code>	<code>ViChar</code>	Returns the interpreted error code as a user readable message string; you must pass a <code>ViChar</code> array at least <code>MAX_ERROR_DESCRIPTION</code> bytes in length.

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning

0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetError

C Function Prototype

```
ViStatus niScope_GetError (ViSession vi, ViStatus* errorCode, ViInt32 bufferSize, ViChar[] description);
```

Purpose



Note This function is included for compliance with the IviScope Class Specification.

Reads an error code and message from the error queue. National Instruments digitizers do not contain an error queue. Errors are reported as they occur. Therefore, this function does not detect errors.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
errorCode	ViStatus*	Pass the Error Code that is returned from any of the instrument driver functions.

Output

Name	Type	Description
bufferSize	ViInt32	Passes the number of bytes in the ViChar array you specify for the Description parameter. If the error description, including the terminating NULL byte, contains more bytes than you indicate in this parameter, the function copies bufferSize - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass `VI_NULL` for the description parameter.

`description ViChar[]` Returns the error description for the IVI session or execution thread. If there is no description, the function returns an empty string. The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter.

If you pass 0 for the `bufferSize`, you can pass `VI_NULL` for this parameter.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetErrorMessage

C Function Prototype

```
ViStatus niScope_GetErrorMessage (ViSession vi, ViStatus errorCode, ViInt32 bufferSize, ViChar errorMessage[]);
```

Purpose

Returns the error code from an NI-SCOPE function as a user-readable string. Use `VI_NULL` as the default instrument handle.

You must call this function twice. For the first call, set `bufferSize` to 0 to prevent the function from populating the error message. Instead, the function returns the size of the error string. Use the returned size to create a buffer, then call the function again,

passing in the new buffer and setting `bufferSize` equal to the size that was returned in the first function call.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>errorCode</code>	<code>ViStatus</code>	The error code that is returned from any of the instrument driver functions.
<code>bufferSize</code>		The number of characters you specify for the <code>errorMessage</code> parameter.
Output		
Name	Type	Description
<code>errorMessage[]</code>	<code>ViChar</code>	Returns a char buffer that will be populated with the error message. It should be at least as large as the buffer size.

Return Value

Name	Type	Description								
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_LockSession

C Function Prototype

```
ViStatus niScope_LockSession (ViSession vi, ViBoolean* callerHasLock);
```


Purpose

Obtains a multithread lock on the instrument session. Before doing so, it waits until all other execution threads have released their locks on the instrument session. Other threads might have obtained a lock on this session in the following ways:

- Your application called `niScope_LockSession`
- A call to the instrument driver locked the session
- A call to the IVI engine locked the session

After your call to `niScope_LockSession` returns successfully, no other threads can access the instrument session until you call `niScope_UnlockSession`. Use `niScope_LockSession` and `niScope_UnlockSession` around a sequence of calls to instrument driver functions if you require that the instrument retain its settings through the end of the sequence.

You can safely make nested calls to `niScope_LockSession` within the same thread. To completely unlock the session, you must balance each call to `niScope_LockSession` with a call to `niScope_UnlockSession`. If, however, you use the `callerHasLock` in all calls to `niScope_LockSession` and `niScope_UnlockSession` within a function, the IVI Library locks the session only once within the function regardless of the number of calls you make to `niScope_LockSession`. This allows you to call `niScope_UnlockSession` just once at the end of the function.

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.

Output

Name	Type	Description
<code>callerHasLock</code>	<code>ViBoolean*</code>	This parameter serves as a convenience. If you do not want to use this parameter, pass <code>VI_NULL</code> . Use this parameter in complex functions to keep track of whether you have obtained a lock and therefore need to unlock the session. Pass

the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to `VI_FALSE`. Pass the address of the same local variable to any other calls you make to [niScope_LockSession](#) or [niScope_UnlockSession](#) in the same function.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_UnlockSession

C Function Prototype

```
ViStatus niScope_UnlockSession (ViSession vi, ViBoolean* callerHasLock);
```

Purpose

Releases a lock that you acquired on an instrument session using [niScope_LockSession](#).

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
Output		
Name	Type	Description

`callerHasLock ViBoolean*` This parameter serves as a convenience; if you do not want to use this parameter, pass `VI_NULL`.
 Use this parameter in complex functions to keep track of whether you have obtained a lock and therefore need to unlock the session; pass the address of a local `ViBoolean` variable; in the declaration of the local variable, initialize it to `VI_FALSE`; pass the address of the same local variable to any other calls you make to `niScope_LockSession` or `niScope_UnlockSession` in the same function.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

IVI Compliance and Obsolete Functions

Expand this book to view the IVI compliance and obsolete functions in NI-SCOPE.
[niScope_ClearError](#)

C Function Prototype

```
ViStatus niScope_ClearError (ViSession vi);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Clears the error information for the current execution thread and the IVI session you specify. If you pass `VI_NULL` for the Instrument Handle parameter, this function clears the error information only for the current execution thread.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:


Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ClearInterchangeWarnings

C Function Prototype

```
ViStatus niScope_ClearInterchangeWarnings (ViSession vi);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Clears the list of current interchange warnings.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureAcquisitionRecord

C Function Prototype

```
ViStatus niScope_ConfigureAcquisitionRecord (ViSession vi,
ViReal64 timeperRecord, ViInt32 minNumPoints, ViReal64 acquisitionStartTime);
```

Purpose

This function is included for compliance with the IviScope Class Specification.

Configures the most commonly configured attributes of the instrument acquisition subsystem.

Parameters

Input

Name	Type	Description
------	------	-------------

vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
timeperRecord	ViReal64	Specifies the time per record. Units: Seconds.
minNumPoints	ViInt32	Pass the minimum number of points you require in the record for each channel. Call niScope_ActualRecordLength to obtain the actual record length used. Valid Values: 1 – available onboard memory
acquisitionStartTime	ViReal64	Specifies the position of the first point in the waveform record relative to the trigger event.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureChannel

C Function Prototype

```
ViStatus niScope_ConfigureChannel (ViSession vi, ViConstString channel, ViReal64 range, ViReal64 offset, ViInt32 coupling, ViReal64 probeAttenuation, ViBoolean enabled);
```

Purpose

This function is included for compliance with the IviScope Class Specification.

Configures the most commonly configured attributes of the instrument's channel subsystem.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax . Default Value: "0"
range	ViReal64	Specifies the voltage range for the specified channel(s).
offset	ViReal64	Selects the DC offset added to the specified channel(s). Default Value: 0
coupling	ViInt32	Specify how you want the digitizer to couple the input signal for the channel. Defined Values NISCOPE_VAL_AC (0) NISCOPE_VAL_DC (1) NISCOPE_VAL_GND (2) A certain amount of delay is required for the coupling capacitor to charge after changing vertical coupling from DC to AC. This delay is typically: Low Impedance Source—150 ms 10X Probe—1.5 s 100X Probe—15 s
probeAttenuation	ViReal64	Specifies the probe attenuation for the specified channel(s). Default Value: 1.00 Valid Range: 1.00 – 100 If you have a probe with yX attenuation, set this parameter to y . For example, enter a value of 10 for a 10X probe.

enabled	ViBoolean	Specify whether to enable the digitizer to acquire data for the channel when you call niScope_InitiateAcquisition or niScope_ReadWaveform . Default Value: NISCOPE_VAL_TRUE (1) Defined Values NISCOPE_VAL_TRUE (1) —Acquire data on this channel NISCOPE_VAL_FALSE (0) —Do not acquire data on this channel
---------	-----------	---

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureEdgeTriggerSource

C Function Prototype

ViStatus niScope_ConfigureEdgeTriggerSource (ViSession vi, ViConstString source, ViReal64 level, ViInt32 slope);

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Sets the edge triggering attributes. An edge trigger occurs when the trigger signal specified with the source parameter passes through the voltage threshold specified with the level parameter and has the slope specified with the slope parameter.

This function affects instrument behavior only if the `triggerType` is `NISCOPE_VAL_EDGE`. Set the trigger type and trigger coupling before calling this function.

If the trigger source is one of the analog input channels, you must configure the vertical range, vertical offset, vertical coupling, probe attenuation, and the maximum input frequency before calling this function.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>level</code>	<code>ViReal64</code>	The voltage threshold for the trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_LEVEL</code> for more information.
<code>level</code>	<code>ViReal64</code>	The voltage threshold for the trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_LEVEL</code> for more information.
<code>slope</code>	<code>ViInt32</code>	Specifies whether you want a rising edge or a falling edge to trigger the digitizer. Refer to <code>NISCOPE_ATTR_TRIGGER_SLOPE</code> for more information.

Return Value

Name	Type	Description
<code>Status</code>	<code>ViStatus</code>	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureRefLevels

C Function Prototype

```
ViStatus niScope_ConfigureRefLevels (ViSession vi, ViReal64 low, ViReal64 mid, ViReal64 high);
```

Purpose

This function is included for compliance with the IviScope Class Specification.

Configures the reference levels for all channels of the digitizer. The levels may be set on a per channel basis by setting NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL, NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL, and NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL.

This function configures the reference levels for waveform measurements. Call this function before calling niScope_FetchMeasurement to take a rise time, fall time, width negative, width positive, duty cycle negative, or duty cycle positive measurement.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from <u>niScope_init</u> that identifies a particular instrument session.
low	ViReal64	Pass the low reference you want the digitizer to use for waveform measurements. Units: Either a percentage or voltage based on <u>NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS</u> . A percentage is calculated with the voltage low and voltage high measurements representing 0% and 100%, respectively. Default Value: 10.0
mid	ViReal64	Pass the mid reference you want the digitizer to use for waveform measurements.

Units: Either a percentage or voltage based on [NISCOPE_ATTR_MEAS_REF_LVL_UNITS](#). A percentage is calculated with the voltage low and voltage high measurements representing 0% and 100%, respectively.

Default Value: 50.0

high ViReal64 Pass the high reference you want the digitizer to use for waveform measurements.

Units: Either a percentage or voltage based on [NISCOPE_ATTR_MEAS_REF_LVL_UNITS](#). A percentage is calculated with the voltage low and voltage high measurements representing 0% and 100%, respectively.

Default Value: 90.0

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTrigger

C Function Prototype

```
ViStatus niScope_ConfigureTrigger (ViSession vi, ViInt32 triggerType, ViReal64 holdoff);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Configures the common attributes of the trigger subsystem.

When you use `niScope_ReadWaveform`, the instrument waits for a trigger. You specify the type of trigger for which the instrument waits with the Trigger Type parameter.

If the instrument requires multiple waveform acquisitions to build a complete waveform, it waits for the length of time you specify with the holdoff parameter to elapse since the previous trigger. The instrument then waits for the next trigger.

Related topics:

- [Configuring Triggers](#)
- [Coercions of Trigger Parameters](#)
- [Triggering](#)
- [NI-SCOPE Programming Flow](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
triggerType	ViInt32	Specifies the type of trigger for which the digitizer will wait.
holdoff	ViReal64	The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to <code>NISCOPE_ATTR_TRIGGER_HOLDOFF</code> for more information.

Return Value

Name	Type	Description						
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call <code>niScope_GetErrorMessage</code> . To obtain additional information concerning the error condition, use <code>niScope_GetError</code> and <code>niScope_ClearError</code> . The general meaning of the status code is as follows:						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings
Value	Meaning							
0	Success							
Positive Values	Warnings							

Negative Values

Errors

niScope_ConfigureTriggerCoupling

C Function Prototype

```
ViStatus niScope_ConfigureTriggerCoupling (ViSession vi, ViInt32 coupling);
```

Purpose



Note This function is included for compliance with the IviScope Class Specification.

Sets the trigger coupling attribute.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
coupling	ViInt32	Specify how you want the instrument to couple the trigger signal.

Defined Values

```
NISCOPE_VAL_AC (0)
NISCOPE_VAL_DC (1)
NISCOPE_VAL_HF_REJECT (2)
NISCOPE_VAL_LF_REJECT (3)
NISCOPE_VAL_AC_PLUS_HF_REJECT (1001)
```

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:


Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTriggerOutput

C Function Prototype

```
ViStatus niScope_ConfigureTriggerOutput (ViSession vi, ViInt32 triggerEvent, ViConstString triggerOutput);
```

Purpose

 **Note** This function is obsolete. Consider using [niScope_ExportSignal](#) instead.

Configures the digitizer to generate a signal pulse that other digitizers can detect when configured for digital triggering.

For NI-DAQmx devices, closing the session clears the route. However, if you want to clear the routes before closing the session, call this function again and route the NISCOPE_VAL_STOP_TRIGGER_EVENT to NISCOPE_VAL_NONE.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
triggerEvent	ViInt32	Specifies the condition in which this device generates a digital pulse.
triggerOutput	ViConstString	Specifies the hardware signal line on which the digital pulse is generated. Valid Values NISCOPE_VAL_NO_EVENT NISCOPE_VAL_STOP_TRIGGER_EVENT NISCOPE_VAL_START_TRIGGER_EVENT

```
NISCOPE_VAL_END_OF_ACQUISITION_EVENT
NISCOPE_VAL_END_OF_RECORD_EVENT
```

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTVTriggerLineNumber

C Function Prototype

```
ViStatus niScope_ConfigureTVTriggerLineNumber (ViSession vi,
ViInt32 lineNumber);
```

Purpose

This function is included for compliance with the IviScope Class Specification.

Configures the TV line upon which the instrument triggers. The line number is absolute and not relative to the field of the TV signal.

This function affects instrument behavior only if the trigger type is set to `NISCOPE_VAL_TV_TRIGGER` and the TV trigger event is set to `NISCOPE_VAL_TV_EVENT_LINE_NUMBER`. Call [niScope_ConfigureTVTriggerSource](#) to set the TV trigger event before calling this function.

Parameters

Input		
Name	Type	Description

vi ViSession The instrument handle you obtain from [niScope_init](#) that identifies a particular instrument session.

lineNumber ViInt32 Specify the line number of the signal you want to trigger off of. The valid ranges of the attribute depend on the signal format configured.
Default Value: 1

Signal Format	Line Numbers
M-NTSC, 480i, 480p	1 to 525
BG/PAL, SECAM, 576i, 576p	1 to 625
720p	1 to 750
1080i,1080p	1 to 1,125

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ConfigureTVTriggerSource

C Function Prototype

```
ViStatus niScope_ConfigureTVTriggerSource (ViSession vi, ViConstString source, ViInt32 signalFormat, ViInt32 event, ViInt32 polarity);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Configures the instrument for TV triggering. It configures the TV signal format, the event, and the signal polarity.

This function affects instrument behavior only if the trigger type is `NISCOPE_VAL_TV_TRIGGER`. Set the trigger type and trigger coupling before calling this function.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>source</code>	<code>ViConstString</code>	Pass the source you want the digitizer to monitor for a trigger. Defined Values "0"—Channel 0 "1"—Channel 1 <code>NISCOPE_VAL_EXTERNAL</code> —Analog External Trigger Input
<code>signalFormat</code>	<code>ViInt32</code>	Specifies the Video/TV signal format. Defined Values <code>NISCOPE_VAL_NTSC</code> (1) <code>NISCOPE_VAL_PAL</code> (2) <code>NISCOPE_VAL_SECAM</code> (3)
<code>event</code>	<code>ViInt32</code>	Video/TV event to trigger off of. Defined Values <code>NISCOPE_VAL_TV_EVENT_FIELD1</code> (1)—trigger on field 1 of the signal <code>NISCOPE_VAL_TV_EVENT_FIELD2</code> (2)—trigger on field 2 of the signal <code>NISCOPE_VAL_TV_EVENT_ANY_FIELD</code> (3)—trigger on the first field acquired <code>NISCOPE_VAL_TV_EVENT_ANY_LINE</code> (4)—trigger on the first line acquired <code>NISCOPE_VAL_TV_EVENT_LINE_NUMBER</code> (5)—trigger on a specific line of a video signal. Valid values vary depending on the signal format configured.
<code>polarity</code>	<code>ViInt32</code>	Specifies the polarity of the video signal to trigger off of.

Defined Values

NISCOPE_VAL_TV_POSITIVE (1)
 NISCOPE_VAL_TV_NEGATIVE (2)

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_error_message

C Function Prototype

```
ViStatus niScope_error_message (ViSession vi, ViStatus errorCode, ViChar
errorMessage[IVI_MAX_MESSAGE_BUF_SIZE]);
```

Purpose

Takes the error code returned by NI-SCOPE functions and returns the interpretation as a user-readable string. You can pass `VI_NULL` as the instrument handle, which is useful to interpret errors after [niScope_init](#) has failed.



Note This function is obsolete. Consider using [niScope_GetErrorMessage](#) instead.

Parameters

Input		
Name	Type	Description

vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
errorCode	ViStatus	The error code that is returned from any of the instrument driver functions.
Output		
Name	Type	Description
errorMessage[]	ViChar	Returns the interpreted error code as a user-readable string; you must pass a ViChar array at least <code>IVI_MAX_MESSAGE_BUF_SIZE</code> bytes in length.

niScope_error_query

C Function Prototype

```
ViStatus niScope_error_query (ViSession vi, ViInt32* errCode, ViChar[] errorMessage);
```

Purpose



Note This function is included for compliance with the IviScope Class Specification.

Reads an error code and message from the error queue. National Instruments digitizers do not contain an error queue. Errors are reported as they occur. Therefore, this function does not detect errors.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
Output		
Name	Type	Description
errCode	ViInt32*	Returns the error code for the session or execution thread. If you pass 0 for the Buffer Size, you can pass <code>VI_NULL</code> for this parameter.
errorMessage	ViChar[]	Formats the error code into a user-readable message string. The array must contain at least 256 elements (<code>ViChar[256]</code>).

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_FetchWaveform

C Function Prototype

```
ViStatus niScope_FetchWaveform (ViSession vi, ViConstString channel,
ViInt32 waveformSize, ViReal64[] waveform, ViInt32* actualPoints, ViReal64* initialX,
ViReal64* xIncrement);
```

Purpose



Note This function is included for compliance with the IviScope Class Specification.

Returns the waveform from a previously initiated acquisition that the digitizer acquires for the channel you specify.

[niScope_InitiateAcquisition](#) starts an acquisition on the channels that you enable with [niScope_ConfigureVertical](#). The digitizer acquires waveforms for the enabled channels concurrently. You use [niScope_AcquisitionStatus](#) to determine when the acquisition is complete. You must call this function separately for each enabled channel to obtain the waveforms.

You can call [niScope_ReadWaveform](#) instead of [niScope_InitiateAcquisition](#). [niScope_ReadWaveform](#) starts an acquisition on all enabled channels, waits for the acquisition to complete, and returns the waveform for the channel you

specify. Call this function to obtain the waveforms for each of the remaining channels.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax . Default Value: "0"
waveformSize	ViInt32	The number of elements to insert into the waveform array.
Output		
Name	Type	Description
waveform	ViReal64[]	Returns the waveform that the digitizer acquires. Units: volts Notes: If the digitizer cannot sample a point in the waveform, this function returns an error.
actualPoints	ViInt32*	Indicates the actual number of points the function placed in the waveform array.
initialX	ViReal64*	Indicates the time of the first point in the waveform array relative to the Reference Position. Units: seconds For example, if the digitizer acquires the first point in the waveform array 1 second before the trigger, this parameter returns the value -1.0. If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0.
xIncrement	ViReal64*	Indicates the length of time between points in the waveform array. Units: seconds

Return Value

Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_FetchWaveformMeasurement

C Function Prototype

```
ViStatus niScope_FetchWaveformMeasurement (ViSession vi, ViConstString channel,
ViInt32 measFunction, ViReal64* measurement);
```

Purpose



Notes This function is included for compliance with the IviScope Class Specification. You can use [niScope_ReadWaveformMeasurement](#) instead of this function. [niScope_ReadWaveformMeasurement](#) starts an acquisition on all enabled channels, waits for the acquisition to complete, obtains a waveform measurement on the specified channel, and returns the waveform for the specified channel. Call this function separately to obtain any other waveform measurements on a specific channel.

Configure the appropriate reference levels before calling this function. You can configure the low, mid, and high references by setting the following attributes:

[NISCOPE_ATTR_MEAS_HIGH_REF](#)

[NISCOPE_ATTR_MEAS_LOW_REF](#)

[NISCOPE_ATTR_MEAS_MID_REF](#)

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax . Default Value: "0"
measFunction	ViInt32	Characteristic of the acquired waveform to be measured.

Output

Name	Type	Description
measurement	ViReal64*	The measured value.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetChannelName

C Function Prototype

```
ViStatus niScope_GetChannelName (ViSession vi, ViInt32 index, ViInt32 bufferSize, ViChar[] channelString);
```

Purpose



Note This function is included for compliance with the IviScope Class Specification.

Returns the channel string that is in the channel table at an index you specify. Not applicable to National Instruments digitizers.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
index	ViInt32	A 1-based index into the channel table.
bufferSize	ViInt32	<p>Passes the number of bytes in the ViChar array you specify for the description parameter.</p> <p>If the error description, including the terminating NULL byte, contains more bytes than you indicate in this parameter, the function copies BufferSize - 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.</p> <p>If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.</p>

Output

Name	Type	Description
channelString	ViChar[]	Returns the channel string that is in the channel table at the index you specify. Do not modify the contents of the channel string.

Return Value

Name	Type	Description
Status	ViStatus	<p>Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage. To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError. The general meaning of the status code is as follows:</p>

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetNextCoercionRecord

C Function Prototype

```
ViStatus niScope_GetNextCoercionRecord (ViSession vi, ViInt32 bufferSize,
ViChar[] record);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Returns the coercion information associated with the IVI session. This function retrieves and clears the oldest instance in which the instrument driver coerced a value you specified to another value.

If you set `NISCOPE_ATTR_RECORD_COERCIONS` to `VI_TRUE`, NI-SCOPE keeps a list of all coercions it makes on `ViInt32` or `ViReal64` values that you pass to instrument driver functions. Use this function to retrieve information from that list.

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
bufferSize	ViInt32	<p>Passes the number of bytes in the <code>ViChar</code> array you specify for the Description parameter.</p> <p>If the error description, including the terminating NULL byte, contains more bytes than you indicate in this parameter, the function copies <code>bufferSize - 1</code> bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the <code>bufferSize</code> is 4, the function places "123" into the buffer and returns 7.</p> <p>If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.</p> <p>If you pass 0, you can pass <code>VI_NULL</code> for the Description buffer parameter.</p>

Output

Name	Type	Description
record	ViChar[]	Returns the next coercion record for the IVI session. If there are no coercion records, the function returns an empty string. The buffer must contain at least as many elements as the value you specify with the bufferSize parameter.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_GetNextInterchangeWarning

C Function Prototype

```
ViStatus niScope_GetNextInterchangeWarning (ViSession vi, ViInt32 bufferSize, ViChar[] interchangeWarning);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Returns the interchangeability warnings associated with the IVI session. It retrieves and clears the oldest instance in which the class driver recorded an interchangeability warning. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior.

Use this function to retrieve interchangeability warnings. The driver performs interchangeability checking when [NISCOPE_ATTR_INTERCHANGE_CHECK](#) is set

to `VI_TRUE`. The function returns an empty string in the `interchangeWarning` parameter if no interchangeability warnings remain for the session.

In general, the instrument driver generates interchangeability warnings when an attribute that affects the behavior of the instrument is in a state that you did not specify.

Parameters

Input		
Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from <code>niScope_init</code> that identifies a particular instrument session.
<code>bufferSize</code>	<code>ViInt32</code>	<p>Passes the number of bytes in the <code>ViChar</code> array you specify for the <code>Description</code> parameter.</p> <p>If the error description, including the terminating NULL byte, contains more bytes than you indicate in this parameter, the function copies <code>bufferSize; - 1</code> bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.</p> <p>If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.</p> <p>If you pass 0, you can pass <code>VI_NULL</code> for the <code>Description</code> buffer parameter.</p>
Output		
Name	Type	Description
<code>interchangeWarning</code>	<code>ViChar[]</code>	Returns the next interchange warning for the IVI session. If there are no interchange warnings, the function returns an empty string. The buffer must contain at least as many elements as the value you specify with the <code>bufferSize</code> parameter.

Return Value

Name	Type	Description
------	------	-------------

Status `ViStatus` Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_IsInvalidWfmElement

C Function Prototype

```
ViStatus niScope_IsInvalidWfmElement (ViSession vi, ViReal64 elementValue, ViBoolean* isInvalid);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Determines whether a value you pass from the waveform array is invalid. After the read and fetch waveform functions execute, each element in the waveform array contains either a voltage or a value indicating that the instrument could not sample a voltage.

Parameters

Input

Name	Type	Description
<code>vi</code>	<code>ViSession</code>	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
<code>elementValue</code>	<code>ViReal64</code>	Pass one of the values from the waveform array returned by the read and fetch waveform functions.

Output

Name	Type	Description
------	------	-------------

isInvalid	ViBoolean*	Returns whether the element value is a valid voltage or a value indicating that the digitizer could not sample a voltage. Return values: VI_TRUE—The element value indicates that the instrument could not sample the voltage. VI_FALSE—The element value is a valid voltage.
-----------	------------	--

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ReadWaveform

C Function Prototype

```
ViStatus niScope_ReadWaveform (ViSession vi, ViConstString channel,
ViInt32 waveformSize, ViInt32 maxtime, ViReal64[] waveform, ViInt32* actualPoints,
ViReal64* initialX, ViReal64* xIncrement);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Initiates an acquisition on the channels that you enable with [niScope_ConfigureVertical](#). This function then waits for the acquisition to complete and returns the waveform for the channel you specify. Call [niScope_FetchWaveform](#) to obtain the waveforms for each of the remaining enabled channels without initiating another acquisition.

Use [niScope_ActualRecordLength](#) to determine the required size for the waveform array.

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax . Default Value: "0"
waveformSize	ViInt32	The number of elements to insert into the waveform array.
maxtime	ViInt32	Pass the maximum length of time in which to allow the read waveform operation to complete. If the operation does not complete within this time interval, the function returns the NISCOPE_ERROR_MAX_TIME_EXCEEDED error code. When this occurs, you can call niScope_Abort to cancel the read waveform operation and return the digitizer to the idle state. Units: milliseconds Other Defined Values NISCOPE_VAL_MAX_TIME_NONE NISCOPE_VAL_MAX_TIME_INFINITE
Output		
Name	Type	Description
waveform	ViReal64[]	Returns the waveform that the digitizer acquires. Units: volts
actualPoints	ViInt32*	Indicates the actual number of points the function placed in the waveform array.
initialX	ViReal64*	Indicates the time of the first point in the waveform array relative to the Reference Position. Units: seconds For example, if the digitizer acquires the first point in the waveform array 1 second before the trigger, this parameter returns the value –

		1.0. If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0.
xIncrement	ViReal64*	Indicates the length of time between points in the waveform array. Units: seconds

Return Value


Name	Type	Description								
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Success</td> </tr> <tr> <td>Positive Values</td> <td>Warnings</td> </tr> <tr> <td>Negative Values</td> <td>Errors</td> </tr> </tbody> </table>	Value	Meaning	0	Success	Positive Values	Warnings	Negative Values	Errors
Value	Meaning									
0	Success									
Positive Values	Warnings									
Negative Values	Errors									

niScope_ReadWaveformMeasurement

C Function Prototype

```
ViStatus niScope_ReadWaveformMeasurement (ViSession vi, ViConstString channel,
ViInt32 measFunction, ViInt32 maxTime, ViReal64* measurement);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

Initiates a new waveform acquisition and returns a specified waveform measurement from a specific channel.

This function initiates an acquisition on the channels that you enable with the [niScope_ConfigureVertical](#) function. It then waits for the acquisition to complete, obtains a waveform measurement on the channel you specify, and returns the measurement value. You specify a particular measurement type, such as rise time, frequency, or voltage peak-to-peak.

You can call the [niScope_FetchWaveformMeasurement](#) function separately to obtain any other waveform measurement on a specific channel without initiating another acquisition.

You must configure the appropriate reference levels before calling this function. Configure the low, mid, and high references by calling [niScope_ConfigureRefLevels](#) or by setting the following attributes:

[NISCOPE_ATTR_MEAS_HIGH_REF](#)

[NISCOPE_ATTR_MEAS_LOW_REF](#)

[NISCOPE_ATTR_MEAS_MID_REF](#)

Parameters

Input		
Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.
channel	ViConstString	The channel to configure. For more information, refer to Channel String Syntax . Default Value: "0"
measFunction	ViInt32	The scalar measurement to perform.
maxTime	ViInt32	Pass the maximum length of time in which to allow the read waveform operation to complete. If the operation does not complete within this time interval, the function returns the NISCOPE_ERROR_MAX_TIME_EXCEEDED error code. When this occurs, you can call niScope_Abort to cancel the read waveform operation and return the digitizer to the idle state. Units: milliseconds
Output		
Name	Type	Description
measurement	ViReal64*	The measured value.

Return Value

Name	Type	Description
------	------	-------------

Status ViStatus Reports the status of this operation. To obtain a text description of the status code, call [niScope_GetErrorMessage](#). To obtain additional information concerning the error condition, use [niScope_GetError](#) and [niScope_ClearError](#). The general meaning of the status code is as follows:


Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_ResetInterchangeCheck

C Function Prototype

```
ViStatus niScope_ResetInterchangeCheck (ViSession vi);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification.

When developing a complex test system that consists of multiple test modules, it is generally a good idea to design the test modules so that they can run in any order. To do so requires ensuring that each test module completely configures the state of each instrument it uses.

If a particular test module does not completely configure the state of an instrument, the state of the instrument depends on the configuration from a previously executed test module.

If you execute the test modules in a different order, the behavior of the instrument and therefore the entire test module is likely to change.

This change in behavior is generally instrument-specific and represents an interchangeability problem. You can use this function to test for such cases. After you call this function, the interchangeability checking algorithms in the specific driver ignore all previous configuration operations.

By calling this function at the beginning of a test module, you can determine whether the test module has dependencies on the operation of previously executed test modules.

This function does not clear the interchangeability warnings from the list of previously recorded interchangeability warnings. If you want to guarantee that [niScope_GetNextInterchangeWarning](#) only returns those interchangeability warnings that are generated after calling this function, you must clear the list of interchangeability warnings.

You can clear the interchangeability warnings list by repeatedly calling [niScope_GetNextInterchangeWarning](#) until no more interchangeability warnings are returned. If you are not interested in the content of those warnings, you can call [niScope_ClearInterchangeWarnings](#).

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

niScope_SendSWTrigger

C Function Prototype

```
ViStatus niScope_SendSWTrigger (ViSession vi);
```

Purpose

 **Note** This function is included for compliance with the IviScope Class Specification. Consider using [niScope_SendSoftwareTriggerEdge](#) instead.

Sends a command to trigger the digitizer. Call this function after you call [niScope_ConfigureTriggerSoftware](#).

Parameters

Input

Name	Type	Description
vi	ViSession	The instrument handle you obtain from niScope_init that identifies a particular instrument session.

Return Value

Name	Type	Description
Status	ViStatus	Reports the status of this operation. To obtain a text description of the status code, call niScope_GetErrorMessage . To obtain additional information concerning the error condition, use niScope_GetError and niScope_ClearError . The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

NI-SCOPE Attributes

Expand this book to view the NI-SCOPE attributes.

NISCOPE_ATTR_CHANNEL_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	niScope_ConfigureVertical

Description

Specifies whether the digitizer acquires a waveform for the channel.

Defined Values

VI_TRUE (1)—Acquire data on this channel.

VI_FALSE (0)— Do not acquire data on this channel.

NISCOPE_ATTR_VERTICAL_RANGE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	Yes	niScope_ConfigureVertical

Description

Specifies the absolute value of the input range for a channel in volts. For example, to acquire a sine wave that spans between -5 and $+5$ V, set this attribute to 10.0 V.

Refer to the **NI High-Speed Digitizers Help** for a list of supported vertical ranges for each device. If the specified range is not supported by a device, the value is coerced up to the next valid range.

Related topics:

- [Coercions of Vertical Parameters](#)

NISCOPE_ATTR_VERTICAL_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niScope_ConfigureVertical

Description

Specifies the location of the center of the range with respect to ground in volts. For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this attribute to 5.0 V.



Note This attribute is not supported by all digitizers. Refer to the [Features Supported by Device](#) for a list of vertical offsets supported for each device.

NISCOPE_ATTR_MAX_INPUT_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niScope_ConfigureChanCharacteristics

Description

Specifies the bandwidth of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB. The units are hertz.

Refer to the **NI High-Speed Digitizers Help** for a list of bandwidths supported for each device.

Defined Values

NISCOPE_VAL_BANDWIDTH_FULL (-1.0)

NISCOPE_VAL_BANDWIDTH_DEVICE_DEFAULT (0.0)

NISCOPE_VAL_20MHZ_BANDWIDTH (20000000.0)

NISCOPE_VAL_100MHZ_BANDWIDTH (100000000.0)

NISCOPE_VAL_20MHZ_MAX_INPUT_FREQUENCY (20000000.0)

NISCOPE_VAL_100MHZ_MAX_INPUT_FREQUENCY (100000000.0)

Related topics:

- [Bandwidth](#)

NISCOPE_ATTR_VERTICAL_COUPLING

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	niScope_ConfigureVertical

Description

Specifies how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

Defined Values

NISCOPE_VAL_AC (0)

NISCOPE_VAL_DC (1)

NISCOPE_VAL_GND (2)

Related topics:

- [Input Coupling](#)

NISCOPE_ATTR_INPUT_IMPEDANCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niScope_ConfigureChanCharacteristics

Description

Specifies the input impedance for the channel in Ohms.

Defined Values

NISCOPE_VAL_50_OHMS (50)

NISCOPE_VAL_1_MEG_OHM (1000000)

Related topics:

- [Impedance and Impedance Matching](#)

NISCOPE_ATTR_PROBE_ATTENUATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	niScope_ConfigureVertical

Description

Specifies the probe attenuation for the input channel. For example, for a 10:1 probe, set this attribute to 10.0.

Valid Values: Any positive real number. Typical values are 1, 10, and 100.

Related topics:

- [Probes and Their Effects](#)

NISCOPE_ATTR_CHANNEL_TERMINAL_CONFIGURATIO

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	N/A

Description

Specifies the terminal configuration for the channel.

Defined Values

NISCOPE_VAL_SINGLE_ENDED (0)

NISCOPE_VAL_UNBALANCED_DIFFERENTIAL (1)

NISCOPE_VAL_DIFFERENTIAL (2)

Related topics:

- [NI 5922 Channel Terminal Configuration](#)

NISCOPE_ATTR_DIGITAL_GAIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Applies gain to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Units: Unitless

Default Value: 1

Valid Values

-1.5 to 1.5

Related topics:

- [NI 5622 Onboard Signal Processing \(OSP\)](#)

NISCOPE_ATTR_DIGITAL_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Applies offset to the specified channel in hardware before any onboard signal processing occurs. The output of the digital gain/offset block is as follows:

(ADC value × digital gain) + digital offset



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Units: Volts

Default Value: 0

Valid Values

±(Vertical Range × 0.4)

Related topics:

- [NI 5622 Onboard Signal Processing \(OSP\)](#)

NISCOPE_ATTR_BANDPASS_FILTER_ENABLED

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables the bandpass filter on the specified channel. For the NI PXIe-5622, set the value to TRUE to enable the IF filtered path 50MHz bandpass filter centered at 187MHz.

Default Value:	VI_FALSE
Valid Values:	VI_TRUE or VI_FALSE

Related topics:

- [Bandwidth](#)

NISCOPE_ATTR_DITHER_ENABLED

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables or disables the analog dither on the device.

Using dither can improve the spectral performance of the device by reducing the effects of [quantization](#). However, adding dither increases the power level to the ADC, so you may need to either decrease the signal level or increase the vertical range.

Default Value:	VI_FALSE
Valid Values:	VI_TRUE or VI_FALSE

NISCOPE_ATTR_FLEX_FIR_ANTIALIAS_FILTER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter. Use this attribute to select from several types of filters to achieve desired filtering characteristics. For most applications, the default value of this property is recommended. The other available filters are useful for optimizing settling time measurements of step responses.

Default Value: NISCOPE_VAL_48_TAP_STANDARD

Defined Values



Note Settling time values refer to the FIR filter only and do not take into account settling time caused by the analog front end. Refer to the **NI PXI-5922 Specifications** for combined digital and analog settling times.

NISCOPE_VAL_48_TAP_STANDARD (0)	<p>This filter is optimized for alias protection and frequency-domain flatness.</p> <ul style="list-style-type: none"> ▪ Alias protection: ranges from 80 dB to 100 dB depending on sample rate ▪ Settling time: within 14 samples from a 50% vertical trigger point ▪ Cutoff frequency: $0.43 \times$ sample rate ▪ Flatness: Ripple ranges from 0.005 dB to 0.120 dB depending of selected sample rate (refer to the NI PXI-5922 Specifications for more information) ▪ Rise time: Approximately $0.75/(\text{sample rate})$
NISCOPE_VAL_48_TAP_HANNING (1)	<p>This filter is optimized for the lowest possible bandwidth for a 48 tap filter and maximizes the signal-to-noise ratio (SNR).</p> <ul style="list-style-type: none"> ▪ Settling time: within 14 samples from a 50% vertical trigger point ▪ Cutoff frequency: $0.030 \times$ sample rate ▪ Flatness: 0 to -3 dB within cutoff frequency ▪ Rise time: $11.6/(\text{sample rate})$
NISCOPE_VAL_16_TAP_HANNING (2)	<p>This filter is optimized for the lowest possible bandwidth for a 16 tap filter and maximizes the SNR.</p>

- Settling time: 6 samples from a 50% vertical trigger point
- Cutoff frequency: $0.08 \times \text{sample rate}$
- Flatness: 0 to -3 dB within cutoff frequency
- Rise time: $7.7/(\text{sample rate})$

NISCOPE_VAL_8_TAP_HANNING (3) This filter is optimized for the lowest possible bandwidth for an 8 tap filter and maximizes the SNR.

- Settling time: 4 samples from a 50% vertical trigger point
- Cutoff frequency: $0.15 \times \text{sample rate}$
- Flatness: 0 to -3 dB within cutoff frequency
- Rise time: $3.9/(\text{sample rate})$

Related topics:

- [Aliasing](#)
- [FIR Filters](#)

NISCOPE_ATTR_HIGH_PASS_FILTER_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the frequency for the highpass filter. The device uses one of the valid values listed below. If an invalid value is specified, no coercion occurs.

Units: Hz

Default Value: 0

Valid Values

(PXIe-5164)

- 0
- 90
- 450

Related topics:

- [Digital Filtering](#)


NISCOPE_ATTR_INTERLEAVING_OFFSET_CORRECTION

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables the interleaving offset correction on the specified channel.

 Note If disabled, warranted specifications are not guaranteed.
Default Value: VI_TRUE
Valid Values: VI_TRUE or VI_FALSE

Related topics:

- [Timed Interleaved Sampling](#)

NISCOPE_ATTR_MIN_SAMPLE_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureHorizontalTiming

Description

Specifies the sampling rate (in Samples/second) for the acquisition. This attribute is invalid when the device is configured to use an external sample clock timebase.

When a DDC is enabled, this attribute specifies the IQ rate. When both the `NISCOPE_ATTR_HORZ_TIME_PER_RECORD` and `NISCOPE_ATTR_MIN_SAMPLE_RATE` are set, the attribute that was set first is ignored.

Valid Values: The combination of sampling rate and minimum record length must allow the digitizer to sample at a valid sampling rate for the acquisition type specified in [niScope_ConfigureAcquisition](#) and not require more memory than the onboard memory module allows.

Related topics:

- [Sample Rate](#)
- [Coercions of Horizontal Parameters](#)

NISCOPE_ATTR_HORZ_MIN_NUM_PTS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition.

[NISCOPE_ATTR_HORZ_RECORD_LENGTH](#) returns the actual record length.

Valid Values: 1 – available onboard memory

Related topics:

- [Coercions of Horizontal Parameters](#)

NISCOPE_ATTR_HORZ_NUM_RECORDS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this attribute to 1 indicates a single record acquisition.

Related topics:

- [Making Multiple-Record Acquisitions](#)

NISCOPE_ATTR_HORZ_RECORD_REF_POSITION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time the [NISCOPE_ATTR_TRIGGER_DELAY_TIME](#) attribute specifies. The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.

Valid Values: 0.0 – 100.0

NISCOPE_ATTR_HORZ_SAMPLE_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	niScope_SampleRate

Description

Returns the effective sample rate using the current configuration. This attribute is only valid after a call to the one of the Configure Horizontal functions.

Units: hertz (Samples / Second)

Related topics:

- [Sample Clock](#)

NISCOPE_ATTR_HORZ_RECORD_LENGTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	niScope_ActualRecordLength

Description

Returns the actual number of points the digitizer acquires for each channel. The value is equal to or greater than the minimum number of points you specify with [NISCOPE_ATTR_HORZ_MIN_NUM_PTS](#).

Allocate a ViReal64 array of this size or greater to pass as the WaveformArray parameter of the Read and Fetch functions. This attribute is only valid after a call to the one of the Configure Horizontal functions. The value is equal to or greater than the minimum number of points you specify in [niScope_ConfigureHorizontalTiming](#) functions.

Related topics:

- [Coercions of Horizontal Parameters](#)

NISCOPE_ATTR_ENABLE_TIME_INTERLEAVED_SAMPLING

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Extends the maximum sample rate on the specified channel for devices that support Time Interleaved Sampling (TIS). TIS enables the device to use multiple ADCs to sample the same waveform at a higher effective real-time rate.

Default Value: VI_FALSE (0)

Defined Values

VI_TRUE (1)—Use multiple interleaved ADCs to acquire data for this channel.

VI_FALSE (0)—Use only this channel's ADC to acquire data for this channel.

Related topics:

- [Time Interleaved Sampling](#)
- [Configuring the Horizontal Settings](#)

NISCOPE_ATTR_HORZ_ENFORCE_REALTIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Indicates whether the digitizer enforces real-time measurements or allows equivalent-time measurements.

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [Sampling Methods](#)
- [Real-Time Sampling](#)
- [Time Interleaved Sampling](#)

NISCOPE_ATTR_ALLOW_MORE_RECORDS_THAN_MEM

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	niScope_ConfigureHorizontalTiming

Description

Indicates whether more records can be configured with [niScope_ConfigureHorizontalTiming](#) than fit in the onboard memory. If this attribute is set to VI_TRUE, it is necessary to fetch records while the acquisition is in progress. Eventually, some of the records are overwritten. An error is returned from the fetch function if you attempt to fetch a record that has been overwritten.

Defined Values

VI_TRUE (1)

VI_FALSE (0)

Related topics:

- [Continuously Acquiring Data](#)

NISCOPE_ATTR_RIS_NUM_AVERAGES

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

The number of averages for each bin in an RIS acquisition. The number of averages times the oversampling factor is the minimum number of real-time acquisitions necessary to reconstruct the RIS waveform. Averaging is useful in RIS because the trigger times are not evenly spaced, so adjacent points in the reconstructed waveform cannot be accurately spaced. By averaging, the errors in both time and voltage are smoothed.

Related topics:

- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

NISCOPE_ATTR_RIS_METHOD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of

[NISCOPE_ATTR_MAX_REAL_TIME_SAMPLING_RATE](#).

Defined Values

NISCOPE_VAL_RIS_EXACT_NUM_AVERAGES	(1) Acquires exactly the specified number of records for each bin in the RIS acquisition. An error is returned from the fetch function if the RIS acquisition does not successfully acquire the specified number of waveforms within the timeout period. You may call the fetch function again to allow more time for the acquisition to finish.
NISCOPE_VAL_RIS_MIN_NUM_AVERAGES	(2) Each RIS sample is the average of at least Min Num Avg randomly distributed points. Any extra points taken are also averaged in.
NISCOPE_VAL_RIS_INCOMPLETE	(3) Returns the RIS waveform after the specified timeout even if it is incomplete. If no waveforms have been acquired in certain bins, these bins have a NaN (when fetching scaled data) or a zero (when fetching binary data). A warning (positive error code) is returned from the fetch function if the RIS acquisition did not finish. The acquisition aborts when data is returned.
NISCOPE_VAL_RIS_LIMITED_BIN_WIDTH	(5) Limits the waveforms in the various bins to be within 200 ps of the center of the bin.

Related topics:

- [Equivalent-Time Sampling and Random Interleaved Sampling](#)
- [Configuring the Horizontal Settings](#)

NISCOPE_ATTR_POLL_INTERVAL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the poll interval in milliseconds to use during RIS acquisitions to check whether the acquisition is complete.

NISCOPE_ATTR_REF_TRIG_TDC_ENABLE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

This attribute controls whether the TDC is used to compute an accurate trigger.

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [TDC](#)

NISCOPE_ATTR_HORZ_TIME_PER_RECORD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the length of time (in seconds) that corresponds to the record length. This attribute is invalid when the device is configured to use an external sample clock timebase. This attribute is also invalid when a DDC is enabled. When both NISCOPE_ATTR_HORZ_TIME_PER_RECORD and

NISCOPE_ATTR_MIN_SAMPLE_RATE are set, the attribute that was set first is ignored.

Units: Seconds

Related topics:

- [Coercions of Horizontal Parameters](#)

NISCOPE_ATTR_ACQUISITION_START_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the length of time from the trigger event to the first point in the waveform record in seconds. If the value is positive, the first point in the waveform record occurs after the trigger event (same as specifying [NISCOPE_ATTR_TRIGGER_DELAY_TIME](#)). If the value is negative, the first point in the waveform record occurs before the trigger event (same as specifying [NISCOPE_ATTR_HORZ_RECORD_REF_POSITION](#)).

NISCOPE_ATTR_TRIGGER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the type of trigger to use.

Defined Values

NISCOPE_VAL_DIGITAL_TRIGGER
 NISCOPE_VAL_EDGE_TRIGGER
 NISCOPE_VAL_GLITCH_TRIGGER
 NISCOPE_VAL_HYSTERESIS_TRIGGER
 NISCOPE_VAL_IMMEDIATE_TRIGGER
 NISCOPE_VAL_RUNT_TRIGGER
 NISCOPE_VAL_SOFTWARE_TRIGGER
 NISCOPE_VAL_TV_TRIGGER
 NISCOPE_VAL_WIDTH_TRIGGER
 NISCOPE_VAL_WINDOW_TRIGGER

Related topics:

- [Trigger Types](#)

NISCOPE_ATTR_TRIGGER_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ConfigureTriggerDigital niScope_ConfigureTriggerEdge niScope_ConfigureTriggerGlitch niScope_ConfigureTriggerHysteresis niScope_ConfigureTriggerRunt niScope_ConfigureTriggerVideo niScope_ConfigureTriggerWidth niScope_ConfigureTriggerWindow

Description

Specifies the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.

Defined Values

"0"

"1"

"2"

"3"

"4"

"5"

"6"

"7"

NISCOPE_VAL_IMMEDIATE ("VAL_IMMEDIATE")

NISCOPE_VAL_EXTERNAL ("VAL_EXTERNAL")

NISCOPE_VAL_SW_TRIG_FUNC ("VAL_SW_TRIG_FUNC")

NISCOPE_VAL_RTSI_0 ("VAL_RTSI_0")

NISCOPE_VAL_RTSI_1 ("VAL_RTSI_1")

NISCOPE_VAL_RTSI_2 ("VAL_RTSI_2")

NISCOPE_VAL_RTSI_3 ("VAL_RTSI_3")

NISCOPE_VAL_RTSI_4 ("VAL_RTSI_4")

NISCOPE_VAL_RTSI_5 ("VAL_RTSI_5")

NISCOPE_VAL_RTSI_6 ("VAL_RTSI_6")

NISCOPE_VAL_PFI_0 ("VAL_PFI_0")

NISCOPE_VAL_PFI_1 ("VAL_PFI_1")
 NISCOPE_VAL_PFI_2 ("VAL_PFI_2")
 NISCOPE_VAL_PFI_3 ("VAL_PFI_3")
 NISCOPE_VAL_PFI_4 ("VAL_PFI_4")
 NISCOPE_VAL_PFI_5 ("VAL_PFI_5")
 NISCOPE_VAL_PFI_6 ("VAL_PFI_6")
 NISCOPE_VAL_PFI_7 ("VAL_PFI_7")
 NISCOPE_VAL_PXI_STAR ("VAL_PXI_STAR")

Related topics:

- [Trigger Types](#)

NISCOPE_ATTR_END_OF_RECORD_TO_ADVANCE_TRIG

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition. This attribute disables sampling and triggering.



Note This attribute is only valid for the NI 5185/5186.

Related topics:

- [Record Advance Holdoff](#)

NISCOPE_ATTR_EXPORTED_REF_TRIGGER_OUTPUT_T

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination export for the Reference (Stop) trigger. Refer to the device specifications document for a list of valid destinations.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PFI_3	("VAL_PFI_3")
NISCOPE_VAL_PFI_4	("VAL_PFI_4")
NISCOPE_VAL_PFI_5	("VAL_PFI_5")
NISCOPE_VAL_PFI_6	("VAL_PFI_6")
NISCOPE_VAL_PFI_7	("VAL_PFI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_REF_TRIGGER_TERMINAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Reference Trigger terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_TRIGGER_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerEdge niScope_ConfigureTriggerGlitch niScope_ConfigureTriggerHysteresis niScope_ConfigureTriggerWidth

Description

Specifies the voltage threshold for the trigger subsystem in volts. This attribute affects instrument behavior only when the [NISCOPE_ATTR_TRIGGER_TYPE](#) is set to [NISCOPE_VAL_EDGE](#), [NISCOPE_VAL_HYSTERESIS](#), or [NISCOPE_VAL_WINDOW](#).

Valid Values: The values of the range and offset parameters in [niScope_ConfigureVertical](#) determine the valid range for the trigger level on the channel you use as the Trigger Source. The value you pass for this parameter must meet the following conditions:

Trigger Level \leq (**Vertical Range**/2) + **Vertical Offset**

Vertical Range \geq (-**Vertical Range**/2) + **Vertical Offset**

Related topics:

- [Edge Triggers](#)
- [Window Triggers](#)

NISCOPE_ATTR_TRIGGER_MODIFIER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Configures whether the device automatically completes an acquisition if a trigger is not received after a period of time.



Note When `NISCOPE_VAL_AUTO` is selected, you may need to modify the timeout on calls to fetch data to ensure that the acquisition does not time out before the auto trigger time elapses.

Defined Values

`NISCOPE_VAL_NO_TRIGGER_MOD` (1)

`NISCOPE_VAL_AUTO` (2)

`NISCOPE_ATTR_TRIGGER_AUTO_TRIGGERED`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

Specifies whether the acquisition was triggered automatically. Auto triggering occurs if the Trigger Modifier property is set to Auto Trigger and no trigger has been received for a certain amount of time.

Defined Values

TRUE

FALSE

NISCOPE_ATTR_TRIGGER_HYSTERESIS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies, in volts, the size of the hysteresis window on either side of the trigger level. The digitizer triggers when the trigger signal passes through the threshold you specify with the level control, has the slope you specify with the slope control, and passes through the hysteresis window that you specify with this attribute.

Valid Values

Positive trigger slope	$0 \leq \text{Hysteresis} \leq \text{Trigger Level} + (\text{Vertical Range} / 2) - \text{Vertical Offset}$
Negative trigger slope	$0 \leq \text{Hysteresis} \leq (\text{Vertical Range} / 2) + \text{Vertical Offset} - \text{Trigger Level}$

Related topics:

- [Hysteresis Triggers](#)

NISCOPE_ATTR_TRIGGER_DELAY_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerDigital niScope_ConfigureTriggerEdge niScope_ConfigureTriggerGlitch niScope_ConfigureTriggerHysteresis

				niScope_ConfigureTriggerRunt ConfigureTriggerSoftware ConfigureTriggerVideo ConfigureTriggerWidth ConfigureTriggerWindow
--	--	--	--	--

Description

Specifies the trigger delay time in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

NISCOPE_ATTR_TRIGGER_HOLDOFF

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerDigital niScope_ConfigureTriggerEdge niScope_ConfigureTriggerGlitch niScope_ConfigureTriggerHysteresis niScope_ConfigureTriggerRunt ConfigureTriggerSoftware ConfigureTriggerVideo ConfigureTriggerWidth ConfigureTriggerWindow

Description

Specifies the length of time (in seconds) the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger. This attribute affects instrument operation only when the digitizer requires multiple acquisitions to build a complete waveform. The digitizer requires multiple waveform acquisitions when it uses equivalent-time sampling or when the digitizer is configured for a multirecord acquisition through a call to [niScope_ConfigureHorizontalTiming](#).

Valid Values: 0.0 – 171.8

Related topics:

- [Trigger Holdoff](#)
- [Trigger Parameters](#)

NISCOPE_ATTR_TRIGGER_SLOPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerDigital niScope_ConfigureTriggerEdge niScope_ConfigureTriggerHysteresis

Description

Specifies whether rising or a falling edge triggers the digitizer. This attribute affects instrument operation only when [NISCOPE_ATTR_TRIGGER_TYPE](#) is set to [NISCOPE_VAL_EDGE_TRIGGER](#), [NISCOPE_VAL_HYSTERESIS_TRIGGER](#), or [NISCOPE_VAL_DIGITAL_TRIGGER](#).

Defined Values

[NISCOPE_VAL_POSITIVE](#)

[NISCOPE_VAL_NEGATIVE](#)

Related topics:

- [Edge Triggers](#)
- [Window Triggers](#)

NISCOPE_ATTR_TRIGGER_COUPLING

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies how the digitizer couples the trigger source. This attribute affects instrument operation only when `NISCOPE_ATTR_TRIGGER_TYPE` is set to `NISCOPE_VAL_EDGE`, `NISCOPE_VAL_HYSTERESIS`, or `NISCOPE_VAL_WINDOW`.

Defined Values

`NISCOPE_VAL_AC` (0)

`NISCOPE_VAL_DC` (1)

`NISCOPE_VAL_HF_REJECT` (2)

`NISCOPE_VAL_LF_REJECT` (3)

`NISCOPE_VAL_AC_PLUS_HF_REJECT` (1001)

Related topics:

- [Input Coupling](#)
- [Trigger Coupling](#)

NISCOPE_ATTR_TRIGGER_IMPEDANCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the input impedance for the external analog trigger channel in Ohms.

Defined Values

`NISCOPE_VAL_50_OHMS` (50)

`NISCOPE_VAL_1_MEG_OHM` (1000000)

Supported Devices

- [PXIe-5160](#)
- [PXIe-5162](#)

Related topics:

- [Impedance and Impedance Matching](#)

NISCOPE_ATTR_START_TO_REF_TRIGGER_HOLDOFF

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

The length of time you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a reference (stop) trigger.

Units: Seconds

Valid Values: 0.0 – 171.8

Related topics:

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)

NISCOPE_ATTR_GLITCH_CONDITION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerGlitch

Description

Specifies whether the oscilloscope triggers on pulses of duration less than or greater than the value specified by the [NISCOPE_ATTR_GLITCH_WIDTH](#) attribute.

Default Value: NISCOPE_VAL_GLITCH_GREATER_THAN

Defined Values

NISCOPE_VAL_GLITCH_GREATER_THAN	(1) The oscilloscope triggers on pulses with a duration longer than the specified glitch width.
NISCOPE_VAL_GLITCH_LESS_THAN	(2) The oscilloscope triggers on pulses with a duration shorter than the specified glitch width.

Related topics:

- [Glitch Triggers](#)

NISCOPE_ATTR_GLITCH_POLARITY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerGlitch

Description

Specifies the polarity of pulses that trigger the oscilloscope for glitch triggering.

Default Value: NISCOPE_VAL_GLITCH_POSITIVE

Defined Values

NISCOPE_VAL_GLITCH_POSITIVE	(1) The oscilloscope triggers on pulses of positive polarity relative to the trigger threshold.
NISCOPE_VAL_GLITCH_NEGATIVE	(2) The oscilloscope triggers on pulses of negative polarity relative to the trigger threshold.
NISCOPE_VAL_GLITCH_EITHER	(3) The oscilloscope triggers on pulses of either positive or negative polarity.

Related topics:

- [Glitch Triggers](#)

NISCOPE_ATTR_GLITCH_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerGlitch

Description

Specifies the glitch duration, in seconds.

The oscilloscope triggers when it detects a pulse of duration either less than or greater than this value depending on the value of the [NISCOPE_ATTR_GLITCH_CONDITION](#) attribute.

Default Value: 0.0

Related topics:

- [Glitch Triggers](#)

NISCOPE_ATTR_RUNT_HIGH_THRESHOLD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerRunt

Description

Specifies the higher of two thresholds, in volts, that bound the vertical range to examine for runt pulses.

The runt threshold that causes the oscilloscope to trigger depends on the runt polarity you select. Refer to the [NISCOPE_ATTR_RUNT_POLARITY](#) attribute for more information.

Default Value: 0.0

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_RUNT_LOW_THRESHOLD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerRunt

Description

Specifies the lower of two thresholds, in volts, that bound the vertical range to examine for runt pulses.

The runt threshold that causes the oscilloscope to trigger depends on the runt polarity you select. Refer to the [NISCOPE_ATTR_RUNT_POLARITY](#) attribute for more information.

Default Value: 0.0

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_RUNT_POLARITY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerRunt

Description

Specifies the polarity of pulses that trigger the oscilloscope for runt triggering.

This attribute determines how the oscilloscope triggers relative to the runt thresholds you set.

When set to `NISCOPE_VAL_RUNT_POSITIVE`, the oscilloscope triggers when the following conditions are met:

- The leading edge of a pulse crosses the `NISCOPE_ATTR_RUNT_LOW_THRESHOLD` in a positive direction;
- The trailing edge of the pulse crosses the `NISCOPE_ATTR_RUNT_LOW_THRESHOLD` in a negative direction; and
- No portion of the pulse crosses the `NISCOPE_ATTR_RUNT_HIGH_THRESHOLD`.

When set to `NISCOPE_VAL_RUNT_NEGATIVE`, the oscilloscope triggers when the following conditions are met:

- The leading edge of a pulse crosses the `NISCOPE_ATTR_RUNT_HIGH_THRESHOLD` in a negative direction;
- The trailing edge of the pulse crosses the `NISCOPE_ATTR_RUNT_HIGH_THRESHOLD` in a positive direction; and
- No portion of the pulse crosses the `NISCOPE_ATTR_RUNT_LOW_THRESHOLD`.

When set to `NISCOPE_VAL_RUNT_EITHER`, the oscilloscope triggers in either case.

Default Value: `NISCOPE_VAL_RUNT_POSITIVE`

Defined Values

<code>NISCOPE_VAL_RUNT_POSITIVE</code>	(1) The oscilloscope triggers on pulses of positive polarity relative to <code>NISCOPE_ATTR_RUNT_LOW_THRESHOLD</code> that do not cross <code>NISCOPE_ATTR_RUNT_HIGH_THRESHOLD</code> .
--	---

NISCOPE_VAL_RUNT_NEGATIVE	(2)	The oscilloscope triggers on pulses of negative polarity relative to NISCOPE_ATTR_RUNT_HIGH_THRESHOLD that do not cross NISCOPE_ATTR_RUNT_LOW_THRESHOLD .
NISCOPE_VAL_RUNT_EITHER	(3)	The oscilloscope triggers on pulses of either positive or negative polarity.

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_RUNT_TIME_CONDITION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies whether runt triggers are time qualified, and if so, how the oscilloscope triggers in relation to the duration range bounded by the [NISCOPE_ATTR_RUNT_TIME_LOW_LIMIT](#) and [NISCOPE_ATTR_RUNT_TIME_HIGH_LIMIT](#) attributes.

Default Value: NISCOPE_VAL_RUNT_TIME_CONDITION_NONE

Defined Values

NISCOPE_VAL_RUNT_TIME_CONDITION_NONE	(0)	Time qualification is disabled. The oscilloscope triggers on runt pulses based solely on the voltage level of the pulses.
NISCOPE_VAL_RUNT_TIME_CONDITION_WITHIN	(1)	The oscilloscope triggers on pulses that, in addition to meeting runt voltage criteria, have a duration within the range bounded by NISCOPE_ATTR_RUNT_TIME_LOW_LIMIT and NISCOPE_ATTR_RUNT_TIME_HIGH_LIMIT .
NISCOPE_VAL_RUNT_TIME_CONDITION_OUTSIDE	(2)	The oscilloscope triggers on pulses that, in addition to meeting runt voltage criteria, have a duration not within the range bounded by NISCOPE_ATTR_RUNT_TIME_LOW_LIMIT and NISCOPE_ATTR_RUNT_TIME_HIGH_LIMIT .

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_RUNT_TIME_HIGH_LIMIT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies, in seconds, the high runt threshold time.

This attribute sets the upper bound on the duration of runt pulses that may trigger the oscilloscope. The [NISCOPE_ATTR_RUNT_TIME_CONDITION](#) attribute determines how the oscilloscope triggers in relation to the runt time limits.

Default Value: 0.0

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_RUNT_TIME_LOW_LIMIT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies, in seconds, the low runt threshold time.

This attribute sets the lower bound on the duration of runt pulses that may trigger the oscilloscope. The [NISCOPE_ATTR_RUNT_TIME_CONDITION](#) attribute determines how the oscilloscope triggers in relation to the runt time limits.

Default Value: 0.0

Related topics:

- [Runt Triggers](#)

NISCOPE_ATTR_WIDTH_CONDITION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerWidth

Description

Specifies whether the oscilloscope triggers on pulses within or outside the duration range bounded by the [NISCOPE_ATTR_WIDTH_LOW_THRESHOLD](#) and [NISCOPE_ATTR_WIDTH_HIGH_THRESHOLD](#) attributes.

Default Value: NISCOPE_VAL_WIDTH_WITHIN

Defined Values

NISCOPE_VAL_WIDTH_WITHIN	(1)	The oscilloscope triggers on pulses with a duration within the range bounded by NISCOPE_ATTR_WIDTH_LOW_THRESHOLD and NISCOPE_ATTR_WIDTH_HIGH_THRESHOLD .
NISCOPE_VAL_WIDTH_OUTSIDE	(2)	The oscilloscope triggers on pulses with a duration not within the range bounded by NISCOPE_ATTR_WIDTH_LOW_THRESHOLD and NISCOPE_ATTR_WIDTH_HIGH_THRESHOLD .

Related topics:

- [Width Triggers](#)

NISCOPE_ATTR_WIDTH_HIGH_THRESHOLD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
-----------	--------	------------	----------	----------------------

ViReal64	R/W	N/A	None	niScope_ConfigureTriggerWidth
----------	-----	-----	------	---

Description

Specifies the high width threshold, in seconds.

This attribute sets the upper bound on the duration range that triggers the oscilloscope. The [NISCOPE_ATTR_WIDTH_CONDITION](#) attribute determines how the oscilloscope triggers in relation to the width thresholds.

Default Value: 0.0

Related topics:

- [Width Triggers](#)

NISCOPE_ATTR_WIDTH_LOW_THRESHOLD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerWidth

Description

Specifies the low width threshold, in seconds.

This attribute sets the lower bound on the duration range that triggers the oscilloscope. The [NISCOPE_ATTR_WIDTH_CONDITION](#) attribute determines how the oscilloscope triggers in relation to the width thresholds.

Default Value: 0.0

Related topics:

- [Width Triggers](#)

NISCOPE_ATTR_WIDTH_POLARITY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerWidth

Description

Specifies the polarity of pulses that trigger the oscilloscope for width triggering.

Default Value: NISCOPE_VAL_POSITIVE

Defined Values

NISCOPE_VAL_WIDTH_POSITIVE	(1)	The oscilloscope triggers on pulses of positive polarity relative to the trigger threshold.
NISCOPE_VAL_WIDTH_NEGATIVE	(2)	The oscilloscope triggers on pulses of negative polarity relative to the trigger threshold.
NISCOPE_VAL_WIDTH_EITHER	(3)	The oscilloscope triggers on pulses of either positive or negative polarity.

Related topics:

- [Width Triggers](#)

NISCOPE_ATTR_TRIGGER_WINDOW_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerWindow

Description

Specifies whether you want a trigger to occur when the signal enters or leaves the window specified by [NISCOPE_ATTR_TRIGGER_WINDOW_LOW_LEVEL](#), or [NISCOPE_ATTR_TRIGGER_WINDOW_HIGH_LEVEL](#).

Defined Values

NISCOPE_VAL_ENTERING_WINDOW (0)

NISCOPE_VAL_LEAVING_WINDOW (1)

Related topics:

- [Window Triggers](#)

[NISCOPE_ATTR_TRIGGER_WINDOW_LOW_LEVEL](#)

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerWindow

Description

The lower voltage threshold you want the digitizer to use for window triggering.

The digitizer triggers when the trigger signal enters or leaves the window you specify with this attribute and [NISCOPE_ATTR_TRIGGER_WINDOW_HIGH_LEVEL](#).

Valid Values: The value you pass for this attribute must meet the following conditions:

Low Trigger Level \leq **Vertical Range**/2 + **Vertical Offset**

Low Trigger Level \geq **(-Vertical Range/2) + Vertical Offset**

Low Trigger Level < **High Trigger Level**

Related topics:

- [Window Triggers](#)

NISCOPE_ATTR_TRIGGER_WINDOW_HIGH_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	niScope_ConfigureTriggerWindow

Description

The upper voltage threshold you want the digitizer to use for window triggering.

The digitizer triggers when the trigger signal enters or leaves the window you specify with [NISCOPE_ATTR_TRIGGER_WINDOW_LOW_LEVEL](#) and this attribute.

Valid Values: The value you pass for this parameter must meet the following conditions:

High Trigger Level \leq **Vertical Range**/2 + **Vertical Offset**

High Trigger Level \geq **(-Vertical Range/2) + Vertical Offset**

High Trigger Level > **Low Trigger Level**

Related topics:

- [Window Triggers](#)

NISCOPE_ATTR_TV_TRIGGER_SIGNAL_FORMAT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerVideo

Description

Specifies the type of video signal.

Defined Values

SDTV

NISCOPE_VAL_NTSC (1)

NISCOPE_VAL_PAL (2)

NISCOPE_VAL_SECAM (3)

NISCOPE_VAL_M_PAL (1001)

EDTV

NISCOPE_VAL_480I_59_94_FIELDS_PER_SECOND (1010)

NISCOPE_VAL_480I_60_FIELDS_PER_SECOND (1011)

NISCOPE_VAL_480P_59_94_FRAMES_PER_SECOND (1015)

NISCOPE_VAL_480P_60_FRAMES_PER_SECOND (1016)

NISCOPE_VAL_576I_50_FIELDS_PER_SECOND (1020)

NISCOPE_VAL_576P_50_FRAMES_PER_SECOND (1025)

HDTV

NISCOPE_VAL_720P_50_FRAMES_PER_SECOND (1031)

NISCOPE_VAL_720P_59_94_FRAMES_PER_SECOND (1032)

NISCOPE_VAL_720P_60_FRAMES_PER_SECOND (1033)

NISCOPE_VAL_1080I_50_FIELDS_PER_SECOND (1040)

NISCOPE_VAL_1080I_59_94_FIELDS_PER_SECOND (1041)

NISCOPE_VAL_1080I_60_FIELDS_PER_SECOND (1042)

NISCOPE_VAL_1080P_24_FRAMES_PER_SECOND (1045)

NISCOPE_ATTR_TV_TRIGGER_LINE_NUMBER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerVideo

Description

Specifies the line on which to trigger, if [NISCOPE_ATTR_TV_TRIGGER_EVENT](#) is set to [NISCOPE_VAL_TV_EVENT_LINE_NUMBER](#). The valid range of the attribute depends on the signal format selected.

SignalFormat	LineNumbers
M-NTSC, 480i, 480p	1 to 525
BG/PAL, SECAM, 576i, 576p	1 to 625
720p	1 to 750
1080i, 1080p	1 to 1,125

NISCOPE_ATTR_TV_TRIGGER_POLARITY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerVideo

Description

Specifies whether the video signal sync is positive or negative.

Defined Values

NISCOPE_VAL_TV_POSITIVE (1)

NISCOPE_VAL_TV_NEGATIVE (2)

NISCOPE_ATTR_TV_TRIGGER_EVENT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureTriggerVideo

Description

Specifies the condition in the video signal that causes the digitizer to trigger.

Defined Values

NISCOPE_VAL_TV_EVENT_FIELD1 (1)

NISCOPE_VAL_TV_EVENT_FIELD2 (2)

NISCOPE_VAL_TV_EVENT_ANY_FIELD (3)

NISCOPE_VAL_TV_EVENT_ANY_LINE (4)

NISCOPE_VAL_TV_EVENT_LINE_NUMBER (5)

NISCOPE_ATTR_ENABLE_DC_RESTORE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	niScope_ConfigureTriggerVideo

Description

Restores the video-triggered data retrieved by the digitizer to the video signal's zero reference point.

Defined Values

VI_TRUE—Enable DC restore.

VI_FALSE—Disable DC restore.

NISCOPE_ATTR_REF_TRIGGER_DETECTOR_LOCATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Indicates which reference trigger detection circuitry to use on the device.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default Value: NISCOPE_VAL_ANALOG_DETECTION_CIRCUIT

Valid Values

NISCOPE_VAL_DDC_OUTPUT—Use the onboard signal processing logic to implement the reference trigger. This option detects trigger conditions by analyzing the processed digital signal.

NISCOPE_VAL_ANALOG_DETECTION_CIRCUIT—Use the hardware analog circuitry to implement the reference trigger. This option detects trigger conditions by analyzing the unprocessed analog signal.

NISCOPE_ATTR_REF_TRIGGER_MINIMUM_QUIET_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the amount of time (in seconds) the trigger circuit must not detect a signal above the [trigger level](#) (or below the trigger level if the trigger slope is negative) before the trigger is armed. This attribute is useful for triggering at the beginning of signal bursts instead of in the middle of signal bursts.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default value: 0

Valid Values

Any value greater than or equal to 0.

Related Topics

[Burst Triggers](#)

NISCOPE_ATTR_INPUT_CLOCK_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ConfigureClock

Description

Specifies the input source for the PLL reference clock.

Defined Values

NISCOPE_VAL_NO_SOURCE	("VAL_NO_SOURCE")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_EXTERNAL	("VAL_EXTERNAL")
NISCOPE_VAL_CLK_IN	("VAL_CLK_IN")

NISCOPE_VAL_PXI_CLOCK ("VAL_PXI_CLOCK")
 NISCOPE_VAL_RTSI_CLOCK ("VAL_RTSI_CLOCK")
 NISCOPE_ATTR_REF_CLK_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

If [NISCOPE_ATTR_INPUT_CLOCK_SOURCE](#) is an external source, this attribute specifies the frequency of the input, or reference clock, to which the internal sample clock timebase is synchronized. The frequency is in hertz.

NISCOPE_ATTR_OUTPUT_CLOCK_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ConfigureClock

Description

Exports the reference clock to the specified output terminal.

Defined Values

NISCOPE_VAL_NO_SOURCE

NISCOPE_VAL_RTSI_CLOCK

NISCOPE_VAL_PFI_0

NISCOPE_VAL_PFI_1

NISCOPE_VAL_PFI_2

NISCOPE_VAL_CLK_OUT

Related topics:

- [Sample Clock](#)
- [NI-TClk Overview](#)

NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source of the sample clock timebase, which is the timebase used to control waveform sampling. The actual sample rate may be the timebase itself or a divided version of the timebase, depending on the [NISCOPE_ATTR_MIN_SAMPLE_RATE](#) attribute (for internal sources) or the [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_DIV](#) and [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_MULT](#) attributes (for external sources).

Defined Values

NISCOPE_VAL_CLK_IN

NISCOPE_VAL_NO_SOURCE

NISCOPE_VAL_PXI_STAR

NISCOPE_VAL_PFI_0

NISCOPE_VAL_PFI_1

VAL_ONBOARD_CONFIGURABLE_RATE_CLK

NISCOPE_ATTR_SAMP_CLK_TIMEBASE_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
-----------	--------	------------	----------	----------------------

ViReal64	R/W	N/A	None	None
----------	-----	-----	------	------

Description

Specifies, in hertz, the frequency of the external clock used as the timebase source if [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC](#) is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.

[NISCOPE_ATTR_SAMP_CLK_TIMEBASE_DIV](#)

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

If [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC](#) is an external source, specifies the ratio between the sample clock timebase rate and the actual sample rate, which can be slower.

[NISCOPE_ATTR_SAMP_CLK_TIMEBASE_MULT](#)

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

If [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC](#) is an external source, this attribute specifies the ratio between the [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_RATE](#) and the actual sample rate, which can be higher. This property can be used in conjunction with the [NISCOPE_ATTR_SAMP_CLK_TIMEBASE_DIV](#) attribute.

Some devices use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample

clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by device and current configuration.

NISCOPE_ATTR_EXPORTED_SAMPLE_CLOCK_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ExportSignal

Description

Exports the sample clock to the specified terminal.

Defined Values

NISCOPE_VAL_NO_SOURCE	("VAL_NO_SOURCE")
NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_CLK_OUT	("VAL_CLK_OUT")

Related topics:

- [Sample Clock](#)
- [NI-TClk Overview](#)
- [Reference Clock/Phase-Lock Loop](#)

NISCOPE_ATTR_EXPORTED_SAMPLE_CLOCK_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ExportSignal

Description

Exports the sample clock to the specified terminal.

Defined Values

NISCOPE_VAL_NO_SOURCE	("VAL_NO_SOURCE")
NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_CLK_OUT	("VAL_CLK_OUT")

Related topics:

- [Sample Clock](#)
- [NI-TClk Overview](#)
- [Reference Clock/Phase-Lock Loop](#)

NISCOPE_ATTR_CLOCK_SYNC_PULSE_SOURCE

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ConfigureClock

Description

Specifies the line on which the sample clock or the one-time sync pulse is sent or received. This line should be the same for all devices to be synchronized.

Defined Values

NISCOPE_VAL_NO_SOURCE

NISCOPE_VAL_RTISI_0

NISCOPE_VAL_RTISI_1

NISCOPE_VAL_RTISI_2

NISCOPE_VAL_RTISI_3

NISCOPE_VAL_RTISI_4

NISCOPE_VAL_RTISI_5

NISCOPE_VAL_RTISI_6

NISCOPE_VAL_PFI_1

NISCOPE_VAL_PFI_2

Related topics:

- [Sample Clock](#)
- [NI-TClk Overview](#)
- [Reference Clock/Phase-Lock Loop](#)

NISCOPE_ATTR_PLL_LOCK_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	N/A	None	None

Description

If TRUE, the PLL has remained locked to the external reference clock since it was last checked. If FALSE, the PLL has become unlocked from the external reference clock since it was last checked.

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [Reference Clock/Phase-Lock Loop](#)

NISCOPE_ATTR_OSCILLATOR_PHASE_DAC_VALUE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Gets or sets the binary phase DAC value that controls the delay added to the phase-locked loop (PLL) of the sample clock.


Valid Values: [0, 65535]

Default Value: " "

Use this attribute when performing manual adjustment to correct for skew and jitter between oscilloscopes synchronized with NI-TClk.

1. Apply time offset with the [niScope_AdjustSampleClockRelativeDelay](#) function.
2. Use this attribute to convert the time offset into a value that can apply the manual adjustment across sessions and improve synchronization repeatability.

For details on performing manual adjustment, refer to [Improving NI-TClk Synchronization of Oscilloscopes with Manual Adjustment](#).

 Note If this value is set, sample clock adjust and TClk cannot do any sub-sample adjustment of the timebase sample clock.

NISCOPE_ATTR_5V_OUT_OUTPUT_TERMINAL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination for the 5 Volt signal. Refer to the device specifications document for a list of valid destinations.

 Note This attribute is supported only for NI 5152/5153/5154 devices.

Defined Values

NISCOPE_VAL_RTISI_0	("VAL_RTISI_0")
NISCOPE_VAL_RTISI_1	("VAL_RTISI_1")
NISCOPE_VAL_RTISI_2	("VAL_RTISI_2")
NISCOPE_VAL_RTISI_3	("VAL_RTISI_3")
NISCOPE_VAL_RTISI_4	("VAL_RTISI_4")
NISCOPE_VAL_RTISI_5	("VAL_RTISI_5")
NISCOPE_VAL_RTISI_6	("VAL_RTISI_6")

NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_MASTER_ENABLE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	niScope_ConfigureClock

Description

Specifies whether the device is a master or a slave. The master typically originates the trigger signal and clock sync pulse. For a standalone device, set this attribute to VI_FALSE.

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [SMC-Based Digitizers Acquisition Engine State Diagram](#)

NISCOPE_ATTR_ACQ_ARM_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Defined Values

NISCOPE_VAL_IMMEDIATE

NISCOPE_VAL_RTSI_0

NISCOPE_VAL_RTSI_1

NISCOPE_VAL_RTSI_2

NISCOPE_VAL_RTSI_3

NISCOPE_VAL_RTSI_4

NISCOPE_VAL_RTSI_5

NISCOPE_VAL_RTSI_6

NISCOPE_VAL_PFI_0

NISCOPE_VAL_PFI_1

NISCOPE_VAL_PFI_2

NISCOPE_VAL_PFI_3

NISCOPE_VAL_PFI_4

NISCOPE_VAL_PFI_5

NISCOPE_VAL_PFI_6

NISCOPE_VAL_PFI_7

NISCOPE_VAL_PXI_STAR

NISCOPE_VAL_SW_TRIG_FUNC

NISCOPE_ATTR_END_OF_ACQUISITION_EVENT_OUTP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records. Refer to the device specifications document for a list of valid destinations.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PFI_3	("VAL_PFI_3")
NISCOPE_VAL_PFI_4	("VAL_PFI_4")
NISCOPE_VAL_PFI_5	("VAL_PFI_5")
NISCOPE_VAL_PFI_6	("VAL_PFI_6")
NISCOPE_VAL_PFI_7	("VAL_PFI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_END_OF_ACQUISITION_EVENT_TERM

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the End of Acquisition Event terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_EXPORTED_START_TRIGGER_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination to export the Start trigger. When the Start trigger is received, the digitizer begins acquiring data. Refer to the device specifications document for a list of valid destinations.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")

```

NISCOPE_VAL_PFI_2 ("VAL_PFI_2")
NISCOPE_VAL_PXI_STAR ("VAL_PXI_STAR")
NISCOPE_ATTR_ARM_REF_TRIG_SRC

```

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the user-configured trigger source.

Defined Values

NISCOPE_VAL_IMMEDIATE

NISCOPE_VAL_PXI_STAR

NISCOPE_VAL_SW_TRIG_FUNC

NISCOPE_VAL_RTSI_0

NISCOPE_VAL_RTSI_1

NISCOPE_VAL_RTSI_2

NISCOPE_VAL_RTSI_3

NISCOPE_VAL_RTSI_4

NISCOPE_VAL_RTSI_5

NISCOPE_VAL_RTSI_6

NISCOPE_VAL_PFI_0

NISCOPE_VAL_PFI_1

NISCOPE_VAL_PFI_2

NISCOPE_VAL_PFI_3

NISCOPE_VAL_PFI_4

NISCOPE_VAL_PFI_5

NISCOPE_VAL_PFI_6

NISCOPE_VAL_PFI_7

NISCOPE_ATTR_EXPORTED_ADVANCE_TRIGGER_OUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Refer to the **NI High-Speed Digitizers Help** for a specific list of valid destinations for your device.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PFI_3	("VAL_PFI_3")
NISCOPE_VAL_PFI_4	("VAL_PFI_4")

NISCOPE_VAL_PFI_5	("VAL_PFI_5")
NISCOPE_VAL_PFI_6	("VAL_PFI_6")
NISCOPE_VAL_PFI_7	("VAL_PFI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_ADV_TRIG_SRC

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Defined Values

NISCOPE_VAL_IMMEDIATE

NISCOPE_VAL_PXI_STAR

NISCOPE_VAL_SW_TRIG_FUNC

NISCOPE_VAL_RTSI_0

NISCOPE_VAL_RTSI_1

NISCOPE_VAL_RTSI_2

NISCOPE_VAL_RTSI_3

NISCOPE_VAL_RTSI_4

NISCOPE_VAL_RTSI_5

NISCOPE_VAL_RTSI_6

NISCOPE_VAL_PFI_0

NISCOPE_VAL_PFI_1

NISCOPE_VAL_PFI_2

NISCOPE_VAL_PFI_3

NISCOPE_VAL_PFI_4

NISCOPE_VAL_PFI_5

NISCOPE_VAL_PFI_6

NISCOPE_VAL_PFI_7

NISCOPE_ATTR_ADVANCE_TRIGGER_TERMINAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Advance Trigger terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_END_OF_RECORD_EVENT_OUTPUT_T

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	None

Description

Specifies the destination for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record. Refer to the device specifications document for a list of valid destinations.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")

NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PFI_3	("VAL_PFI_3")
NISCOPE_VAL_PFI_4	("VAL_PFI_4")
NISCOPE_VAL_PFI_5	("VAL_PFI_5")
NISCOPE_VAL_PFI_6	("VAL_PFI_6")
NISCOPE_VAL_PFI_7	("VAL_PFI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_END_OF_RECORD_EVENT_TERMINAL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the End of Record Event terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_READY_FOR_ADVANCE_EVENT_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ExportSignal

Description

Specifies the destination for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.

Refer to the device-specific documentation in the **NI High-Speed Digitizers Help** for a list of valid destinations for your device.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")
NISCOPE_ATTR_READY_FOR_ADVANCE_EVENT_TERMINAL	

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Ready for Advance Event terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_READY_FOR_START_EVENT_OUTPUT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ExportSignal

Description

Specifies the destination for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.

Refer to the **NI High-Speed Digitizers Help** for a specific list of valid destinations for your device.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")
NISCOPE_VAL_PFI_2	("VAL_PFI_2")
NISCOPE_VAL_PFI_3	("VAL_PFI_3")
NISCOPE_VAL_PFI_4	("VAL_PFI_4")
NISCOPE_VAL_PFI_5	("VAL_PFI_5")
NISCOPE_VAL_PFI_6	("VAL_PFI_6")
NISCOPE_VAL_PFI_7	("VAL_PFI_7")
NISCOPE_VAL_PXI_STAR	("VAL_PXI_STAR")

NISCOPE_ATTR_READY_FOR_START_EVENT_TERMINA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Ready for Start Event terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_READY_FOR_REF_EVENT_OUTPUT_TE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	niScope_ExportSignal

Description

Specifies the destination for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.

Defined Values

NISCOPE_VAL_RTSI_0	("VAL_RTSI_0")
NISCOPE_VAL_RTSI_1	("VAL_RTSI_1")
NISCOPE_VAL_RTSI_2	("VAL_RTSI_2")
NISCOPE_VAL_RTSI_3	("VAL_RTSI_3")
NISCOPE_VAL_RTSI_4	("VAL_RTSI_4")
NISCOPE_VAL_RTSI_5	("VAL_RTSI_5")
NISCOPE_VAL_RTSI_6	("VAL_RTSI_6")
NISCOPE_VAL_PFI_0	("VAL_PFI_0")
NISCOPE_VAL_PFI_1	("VAL_PFI_1")

NISCOPE_VAL_PFI_2 ("VAL_PFI_2")

NISCOPE_VAL_PXI_STAR ("VAL_PXI_STAR")

NISCOPE_ATTR_READY_FOR_REF_EVENT_TERMINAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Ready for Reference Event terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_START_TRIGGER_TERMINAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R	N/A	None	N/A

Description

Returns the fully qualified name for the Start Trigger terminal.

You can use this terminal as the source for a trigger.

NISCOPE_ATTR_BINARY_SAMPLE_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Indicates the bit width of the binary data in the acquired waveform, which is useful for determining which Binary Fetch function to use. Compare to [NISCOPE_ATTR_RESOLUTION](#).

To configure the device to store samples with a lower resolution than the native resolution, set this attribute to the desired binary width. This process can be useful for streaming at faster speeds at the cost of resolution. The least significant bits are lost with this configuration.

Defined Values

8, 16, 32

NISCOPE_ATTR_RESOLUTION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Indicates the bit width of valid data (as opposed to padding bits) in the acquired waveform. Compare to [NISCOPE_ATTR_BINARY_SAMPLE_WIDTH](#).

Related topics:

- [Resolution](#)

NISCOPE_ATTR_ACQUISITION_TYPE

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	niScope_ConfigureAcquisition

Description

Specifies how the oscilloscope acquires data and fills the waveform record.

Defined Values

NISCOPE_VAL_NORMAL (0)	Sets the oscilloscope to normal resolution mode. The oscilloscope can use real-time sampling or equivalent-time sampling.
NISCOPE_VAL_FLEXRES (1001)	Sets legacy oscilloscopes to flexible resolution mode, if supported.
NISCOPE_VAL_DDC (1002)	Sets legacy oscilloscopes to DDC mode, if supported. To use DDC mode for the PXI/PCI-5142 or PXIe-5622, set this attribute to NISCOPE_VAL_NORMAL and set the NISCOPE_ATTR_DDC_ENABLE attribute to VI_TRUE.

Related topics:

- [Acquisition Functions](#)

NISCOPE_ATTR_SAMPLE_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
VIInt32	RO	N/A	None	niScope_SampleMode

Description

Specifies the sample mode the digitizer is currently using.

Defined Values

NISCOPE_VAL_REAL_TIME (0)

NISCOPE_VAL_EQUIVALENT_TIME (1)

NISCOPE_ATTR_FETCH_RELATIVE_TO

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Position to start fetching within one record.

Default Value: NISCOPE_VAL_PRETRIGGER

Defined Values

NISCOPE_VAL_PRETRIGGER (477)—Fetches relative to the first pretrigger point requested with [niScope_ConfigureHorizontalTiming](#).

NISCOPE_VAL_NOW (481)—Fetch data at the last sample acquired.

NISCOPE_VAL_START (482)—Fetch data starting at the first point sampled by the digitizer.

NISCOPE_VAL_TRIGGER (483)—Fetch at the first posttrigger sample.

NISCOPE_VAL_READ_POINTER (388)—The read pointer is set to zero when a new acquisition is initiated. After every fetch the read pointer is incremented to be the sample after the last sample retrieved. Therefore, you can repeatedly fetch relative to the read pointer for a continuous acquisition program.

Related topics:

- [Fetching the Most Recent Data](#)

NISCOPE_ATTR_FETCH_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
-----------	--------	------------	----------	----------------------

ViInt32	R/W	N/A	None	None
---------	-----	-----	------	------

Description

Offset in samples to start fetching data within each record. The offset is applied relative to [NISCOPE_ATTR_FETCH_RELATIVE_TO](#). The offset can be positive or negative.

Default Value: 0

Related topics:

- [Fetching the Most Recent Data](#)

NISCOPE_ATTR_FETCH_RECORD_NUMBER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Zero-based index of the first record to fetch. Use [NISCOPE_ATTR_FETCH_NUM_RECORDS](#) to set the number of records to fetch.

Default Value: 0

Related topics:

- [Making Multiple-Record Acquisitions](#)
- [Fetching Multiple-Record Acquisitions](#)

NISCOPE_ATTR_FETCH_NUM_RECORDS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Number of records to fetch. Use -1 to fetch all configured records.

Default Value: -1

`NISCOPE_ATTR_FETCH_MEAS_NUM_SAMPLES`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.

Default Value: -1

Related topics:

- [Fetching Data](#)
- [Acquiring Data Continuously](#)

`NISCOPE_ATTR_POINTS_DONE`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Actual number of samples acquired since the last fetch, relative to the configured value for [NISCOPE_ATTR_FETCH_RELATIVE_TO](#), including [NISCOPE_ATTR_FETCH_OFFSET](#), and for the current configured [NISCOPE_ATTR_FETCH_RECORD_NUMBER](#).

NISCOPE_ATTR_RECORDS_DONE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Specifies the number of records that have been completely acquired.

Related topics:

- [Making Multiple-Record Acquisitions](#)
- [Fetching Multiple-Record Acquisitions](#)

NISCOPE_ATTR_DATA_TRANSFER_BLOCK_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the maximum number of samples to transfer at one time from the device to host memory. Increasing this number should result in better fetching performance because the driver does not need to restart the transfers as often. However, increasing this number may also increase the amount of page-locked memory required from the system.

Related topics:

- [Fetching Data](#)
- [Acquiring Data Continuously](#)

NISCOPE_ATTR_BACKLOG

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the number of samples ([NISCOPE_ATTR_POINTS_DONE](#)) that have been acquired but not fetched for the record specified by [NISCOPE_ATTR_FETCH_RECORD_NUMBER](#).

NISCOPE_ATTR_ENABLED_CHANNELS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns a comma-separated list of the channels enabled for the session in ascending order.

- If no channels are enabled, this attribute returns an empty string, "".
- If all channels are enabled, this attribute enumerates all of the channels.

Because this attribute returns channels in ascending order, but the order in which you specify channels for the `channelList` input to NI-SCOPE functions is important, the value of this attribute may not necessarily reflect the order in which NI-SCOPE performs certain actions. Refer to [Channel String Syntax](#) for more information on the effects of channel order in NI-SCOPE.

NISCOPE_ATTR_RIS_IN_AUTO_SETUP_ENABLE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	niScope_AutoSetup

Description

Indicates whether the digitizer should use RIS sample rates when searching for a frequency in `autosetup`.

NISCOPE_ATTR_DATA_TRANSFER_MAXIMUM_BANDWIDTH

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	None

Description

Specifies the maximum bandwidth that the device is allowed to consume. The NI device limits itself to transfer fewer bytes per second on the PCIe bus than the value you specify for this property.

NISCOPE_ATTR_DATA_TRANSFER_PREFERRED_PACKET_SIZE

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Specifies the preferred size of the data field in the PCI Express packet. In general, the larger the packet size, the more efficiently the device uses the bus. However, some systems, because of their implementation, perform better with smaller packet sizes. The value of this property must be a power of two (64, 128, ... , 512).

NISCOPE_ATTR_MEAS_OTHER_CHANNEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Channel	None	None

Description

Specifies the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.

Default Value: 0

NISCOPE_ATTR_MEAS_ARRAY_GAIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Every element of an array is multiplied by this scalar value during the Array Gain measurement.

Default Value: 1.0

NISCOPE_ATTR_MEAS_ARRAY_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Every element of an array is added to this scalar value during the Array Offset measurement.

Default Value: 0.0

NISCOPE_ATTR_MEAS_HYSTERESIS_PERCENT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Digital hysteresis that is used in several of the scalar waveform measurements. This attribute specifies the percentage of the full-scale vertical range for the hysteresis window size.

Default Value: 2%

NISCOPE_ATTR_MEAS_LAST_ACQ_HISTOGRAM_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.

Default Value: 256

NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Determines the multiple acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after clearing the measurement history with the function [niScope_ClearWaveformMeasurementStats](#).

Default Value: 256

NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_LOW_VO

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the lowest voltage value included in the multiple-acquisition voltage histogram.

Default Value: -10.0 V

NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM_HIGH_VO

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the highest voltage value included in the multiple acquisition voltage histogram.

Default Value: 10.0 V

NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Determines the multiple acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after clearing the measurement history with [niScope_ClearWaveformMeasurementStats](#).

Default Value: 256

NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOW_VOLTS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the lowest voltage value included in the multiple acquisition time histogram.

Default Value: -10.0 V

NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIGH_VOLT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the highest voltage value included in the multiple-acquisition time histogram.

Default Value: 10.0 V

NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOW_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the lowest time value (in seconds) included in the multiple-acquisition time histogram.

Default Value: $-5.0e-4$ seconds

NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIGH_TIME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the highest time value included in the multiple acquisition time histogram. The units are always seconds.

Default Value: 5.0e-4 seconds

NISCOPE_ATTR_MEAS_POLYNOMIAL_INTERPOLATION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 is linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.

Default Value: 1

NISCOPE_ATTR_MEAS_INTERPOLATION_SAMPLING_FACTOR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Indirectly specifies the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.

Default Value: 2.0

NISCOPE_ATTR_MEAS_FILTER_TYPE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the type of filter, for both IIR and FIR filters.

Default Value: Lowpass

Defined Values

NISCOPE_VAL_MEAS_LOWPASS (0)

NISCOPE_VAL_MEAS_HIGHPASS (1)

NISCOPE_VAL_MEAS_BANDPASS (2)

NISCOPE_VAL_MEAS_BANDSTOP (3)

Related topics:

- [Reference Levels](#)
- [Measuring Reference-Level Crossings](#)

NISCOPE_ATTR_MEAS_FILTER_TAPS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Defines the number of taps (coefficients) for an FIR filter.

Default Value: 25

Related topics:

- [Reference Levels](#)
- [Measuring Reference-Level Crossings](#)

NISCOPE_ATTR_MEAS_FIR_FILTER_WINDOW

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.

Default Value: NISCOPE_VAL_NONE

Defined Values

NISCOPE_VAL_NONE (0)

NISCOPE_VAL_HANNING_WINDOW (409)

NISCOPE_VAL_FLAT_TOP_WINDOW (410)

NISCOPE_VAL_HAMMING_WINDOW (420)

NISCOPE_VAL_TRIANGLE_WINDOW (423)

NISCOPE_VAL_BLACKMAN_WINDOW (424)

Related topics:

- [Reference Levels](#)
- [Measuring Reference-Level Crossings](#)

NISCOPE_ATTR_MEAS_FILTER_WIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at [NISCOPE_ATTR_MEAS_FILTER_CENTER_FREQ](#) \pm one-half width.

Default Value: 1.0e3 Hz

NISCOPE_ATTR_MEAS_FILTER_ORDER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the order of an IIR filter. All positive integers are valid.

Default Value: 2

Related topics:

- [Reference Levels](#)
- [Measuring Reference-Level Crossings](#)

NISCOPE_ATTR_MEAS_FILTER_CUTOFF_FREQ

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
-----------	--------	------------	----------	----------------------

ViReal64	R/W	Channel	None	None
----------	-----	---------	------	------

Description

Specifies the cutoff frequency in hertz for filters of type lowpass and highpass. The cutoff frequency definition varies depending on the filter.

Default Value: 1.0e6 Hz

`NISCOPE_ATTR_MEAS_FILTER_CENTER_FREQ`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

The center frequency in hertz for bandpass and bandstop filters. The width of the filter is specified by `NISCOPE_ATTR_MEAS_FILTER_WIDTH`, where the cutoff frequencies are the center \pm width.

Default Value: 1.0e6 Hz

`NISCOPE_ATTR_MEAS_FILTER_RIPPLE`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the amount of passband ripple in decibels (positive values) for Chebyshev filters.

Default Value: 0.1 dB

NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WAVEFORM

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

The percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This allows eliminating the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.

Default Value: 20.0%

NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Stores the low reference level used in many scalar measurements. Different channels may have different reference levels.

Default Value: 10%

NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Stores the mid reference level used in many scalar measurements. Different channels may have different reference levels.

Default Value: 50%

NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Stores the high reference level used in many scalar measurements. Different channels may have different reference levels.

Default Value: 90%

NISCOPE_ATTR_DDC_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables/disables the [DDC](#) block of the digitizer. When the DDC block is disabled, all DDC-related properties are disabled and have no effect on the acquired signal.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP

Default Value: VI_FALSE

Defined Values

VI_TRUE

VI_FALSE



Tip To achieve maximum flatness in the filter response of the device in DDC mode, set the maximum input frequency to full bandwidth (-1). However, keep in mind that this action does not protect against aliasing frequency content of the input signal above ($0.5 \times$ **Sample Clock Timebase Rate**). When using internal clocking, the Sample Clock Timebase Rate is 100 MS/s.

Related Topics

[DDC Enabled Overview](#)

NISCOPE_ATTR_DDC_DATA_PROCESSING_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

The way in which data is processed by the [DDC](#) block.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default Value: Complex

Defined Values

NISCOPE_VAL_REAL—The waveform data points are real numbers (I data)

NISCOPE_VAL_COMPLEX—The waveform data points are complex numbers (IQ data)

NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_ENA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables/disables frequency translating the data around the user-selected center frequency down to baseband.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default Value: VI_TRUE

Defined Values

VI_TRUE

VI_FALSE

NISCOPE_ATTR_DDC_CENTER_FREQUENCY

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

The frequency at which the [DDC](#) block frequency translates the input data.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Units: Hz

Default Value: 10 MHz

Valid Values

0 – $(0.5 \times \text{Sample Clock Timebase Rate for digitizer})$

NISCOPE_ATTR_FETCH_INTERLEAVED_IQ_DATA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms. If set to TRUE, the number of elements returned by scalar fetch types (such as 16-bit integer) is twice the requested number of samples. If set to FALSE during DDC acquisitions in Complex mode, two noninterleaved arrays of data are returned per channel, per record.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default Value: VI_TRUE

Defined Values

VI_TRUE—A scalar fetch returns an array of waveforms in the following format:
IQIQI...

VI_FALSE—A scalar fetch returns an array of waveforms in the following format:
III...QQQ...

Related topics:

- [Frequency Domain Fundamentals](#)

NISCOPE_ATTR_DDC_Q_SOURCE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	Channel	None	None

Description

Specifies the channel that is the input to the Q data stream of the [DDC](#).



Note For multi-instrument sessions, the channel you define with this attribute as the Q data stream and the channel for the corresponding I data stream must both be located within the same physical instrument.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Valid Values: All valid channels for the device

Default Value: The channel to which the attribute is registered

Frequency Translation

NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_


NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

The I oscillator phase in degrees at the first point acquired.

 **Note** This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Units: Degrees

Default Value: 0

Valid Values

–360 to 360

NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_


NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATION_PHA

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

The Q oscillator phase in degrees at the first point acquired. Use this attribute only when [NISCOPE_ATTR_DDC_DATA_PROCESSING_MODE](#) is set to Complex.

 **Note** This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Units: Degrees

Default Value: 90

Valid Values

–360 to 360

NISCOPE_ATTR_EQUALIZATION_FILTER_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	Channel	None	None

Description

Enables the onboard signal processing equalization FIR block, which is connected directly to the input signal. The equalization filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default value: VI_FALSE

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [Frequency Domain Fundamentals](#)
- [Flatness](#)

NISCOPE_ATTR_EQUALIZATION_NUM_COEFFICIENTS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
-----------	--------	------------	----------	----------------------

ViInt32	RO	Channel	None	None
---------	----	---------	------	------

Description

Returns the number of coefficients that the equalization FIR filter can accept. This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter, any coefficients are valid. Coefficient values should be between +1 and -1.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Related topics:

- [Frequency Domain Fundamentals](#)
- [Flatness](#)

NISCOPE_ATTR_FRACTIONAL_RESAMPLE_ENABLED

Specific Attribute

Datatype	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Enables the onboard signal processing block that resamples the input waveform to the desired sample rate.

Default Value:	VI_FALSE
Valid Values:	VI_TRUE or VI_FALSE

Related topics:

- [Frequency Domain Fundamentals](#)

NISCOPE_ATTR_OVERFLOW_ERROR_REPORTING

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	None

Description

Configures error reporting when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.



Note This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE returns an error if you use this attribute with a device that does not support OSP.

Default Value: Warning

Valid Values

NISCOPE_VAL_ERROR_REPORTING_ERROR (0)—NI-SCOPE returns an error when an overflow has occurred in the OSP block.

NISCOPE_VAL_ERROR_REPORTING_WARNING (1)—NI-SCOPE returns a warning when an overflow has occurred in the OSP block.

NISCOPE_VAL_ERROR_REPORTING_DISABLED (2)—NI-SCOPE does not return an error when an overflow has occurred in the OSP block.


NISCOPE_ATTR_P2P_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
Boolean	R/W	Endpoint	None	None

Description

Specifies whether the digitizer writes data to the peer-to-peer endpoint.

 **Note** This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Default Value: FALSE

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_CHANNELS_TO_STREAM


Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
String	R/W	Endpoint	None	None

Description

Specifies which channels are written to a peer-to-peer endpoint. If multiple channels are specified, the channels are interleaved by sample.

 **Note** This attribute can only be used with high-speed digitizers that support peer-to-peer streaming.

 **Note** This attribute must either be unused or set to all enabled channels on NI 5160/5162 digitizers.

Default Value: 0

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_ENDPOINT_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Endpoint	None	None

Description

Returns the size in samples of the peer-to-peer endpoint.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_SAMPLES_AVAIL_IN_ENDPOINT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Endpoint	None	None

Description

Returns the current number of samples available to stream from a peer-to-peer endpoint.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_MOST_SAMPLES_AVAIL_IN_ENDPOINT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Endpoint	None	None

Description

Returns the most number of samples available to stream from a peer-to-peer endpoint since the last time this attribute was read.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	RO	Endpoint	None	None

Description

Returns the number of samples transferred through the peer-to-peer endpoint since it was last reset.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)


NISCOPE_ATTR_P2P_ENDPOINT_OVERFLOW

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
Boolean	RO	Endpoint	None	None

Description

Returns TRUE if the endpoint FIFO has overflowed. Reset the endpoint to clear the overflow condition.

 Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.


NISCOPE_ATTR_P2P_FIFO_ENDPOINT_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the number of FIFO-based peer-to-peer endpoints this device supports.

 Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)


NISCOPE_ATTR_P2P_ONBOARD_MEMORY_ENABLED


Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
Boolean	R/W	N/A	None	None

Description

Specifies whether the digitizer writes data to onboard memory when a peer-to-peer endpoint is enabled.

 **Note** This attribute can only be used with high-speed digitizers that support peer-to-peer streaming.

 **Note** This attribute is not supported on NI 5160/5162 digitizers.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_STREAM_RELATIVE_TO

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RW	Endpoint	None	None

Description

Determines which trigger peer-to-peer data is streamed relative to.

 **Note** On the NI 5122/5622, only NISCOPE_VAL_STREAM_RELATIVE_TO_START_TRIGGER is valid for this attribute.

Default Value: NISCOPE_VAL_STREAM_RELATIVE_TO_START_TRIGGER (0)

Defined Values

NISCOPE_VAL_STREAM_RELATIVE_TO_START_TRIGGER (0)

NISCOPE_VAL_STREAM_RELATIVE_TO_REFERENCE_TRIGGER (1)

NISCOPE_VAL_STREAM_RELATIVE_TO_SYNC_TRIGGER (2)


NISCOPE_ATTR_P2P_SAMPLES_TRANSFERRED_PER_RECORD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	Endpoint	None	None

Description

Returns the number of samples transferred per record when you set the [NISCOPE_ATTR_P2P_STREAM_RELATIVE_TO](#) attribute to [NISCOPE_VAL_STREAM_RELATIVE_TO_REFERENCE_TRIGGER](#) or [NISCOPE_VAL_STREAM_RELATIVE_TO_SYNC_TRIGGER](#).

 Note This attribute is only supported on NI 5160/5162 digitizers.


NISCOPE_ATTR_P2P_MANUAL_CONFIGURATION_ENABLED

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
Boolean	R/W	Endpoint	None	None

Description

Enables/disables the advanced attributes for a peer-to-peer endpoint. These attributes cannot be used if an endpoint is being configured by NI-P2P, or a resource reservation error will occur.

 Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR

NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	RO	Endpoint	None	None

Description

Returns the address of a hardware register used to grant permission for the peer-to-peer endpoint to write data to another peer. The type of this address is determined by the [NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR_TYPE](#) attribute. Permission is granted in bytes and the register is additive.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR

NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Endpoint	None	None

Description

Specifies the type of address returned from the [NISCOPE_ATTR_P2P_DATA_TRANS_PERMISSION_ADDR](#) attribute.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Default Value: NISCOPE_VAL_ADDR_VIRTUAL

Defined Values

NISCOPE_VAL_ADDR_PHYSICAL (0)

NISCOPE_VAL_ADDR_VIRTUAL (1)

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	R/W	Endpoint	None	None

Description

Specifies the destination for data written by the peer-to-peer endpoint. The type of this address is specified by the [NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_TYPE](#) attribute.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Valid Values

A valid, non-NULL physical or virtual address.

Related topics:

- Peer-to-Peer Streaming

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Endpoint	None	None

Description

Specifies the type of the [NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR](#) attribute.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Default Value: NISCOPE_VAL_ADDR_VIRTUAL

Defined Values

NISCOPE_VAL_ADDR_PHYSICAL (0)

NISCOPE_VAL_ADDR_VIRTUAL (1)

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_SIZE

NISCOPE_ATTR_P2P_DESTINATION_WINDOW_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	R/W	Endpoint	None	None

Description

Specifies the size, in bytes, of the destination window determined by the [NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR](#) and the [NISCOPE_ATTR_P2P_DESTINATION_WINDOW_ADDR_TYPE](#) attributes.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Valid Values

Any non-NULL value.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON
 NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Endpoint	None	None

Description

Specifies the event to push the [NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VALUE](#) attribute to the [NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR](#) attribute. Setting this attribute to NISCOPE_VAL_NOTIFY_DONE pushes the message when the acquisition has completed.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Default Value: NISCOPE_VAL_NOTIFY_NEVER

Defined Values

NISCOPE_VAL_NOTIFY_NEVER (0)

NISCOPE_VAL_NOTIFY_DONE (1)

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADD

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	R/W	Endpoint	None	None

Description

Specifies the address to Push Message push Value to on the event specified by the [NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON](#) attribute.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADD

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Endpoint	None	None

Description

Specifies the type of the [NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR](#) attribute.



Note This attribute can be used only with high-speed digitizers that support peer-to-peer streaming.

Default Value: NISCOPE_VAL_ADDR_VIRTUAL

Defined Values

NISCOPE_VAL_ADDR_PHYSICAL (0)

NISCOPE_VAL_ADDR_VIRTUAL (1)

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VAL

NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_VALUE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt64	R/W	Endpoint	None	None

Description

Specifies the value to be pushed to the [NISCOPE_ATTR_P2P_NOTIFY_MESSAGE_PUSH_ADDR](#) attribute on the event specified by the [NISCOPE_ATTR_P2P_NOTIFY_PUSH_MESSAGE_ON](#) attribute.



Note This attribute is only supported by devices that support peer-to-peer streaming.

Related topics:

- [Peer-to-Peer Streaming](#)

NISCOPE_ATTR_DEVICE_TEMPERATURE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the temperature of the device in degrees Celsius from the onboard sensor.

Related topics:

- [Calibration](#)
- [Thermal Shutdown](#)
- [PXI/PXIe Chassis Cooling](#)

NISCOPE_ATTR_SERIAL_NUMBER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Returns the serial number of the device.


NISCOPE_ATTR_SIGNAL_COND_GAIN

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Channel	Yes	None

Description

Returns the calibration gain for the current device configuration.

 Note This attribute is supported only by the NI PXI-5900 differential amplifier.

NISCOPE_ATTR_SIGNAL_COND_OFFSET

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	Channel	Yes	None

Description

Returns the calibration offset for the current device configuration.

 Note This attribute is supported only by the NI PXI-5900 differential amplifier.


NISCOPE_ATTR_CABLE_SENSE_MODE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R	N/A	None	None

Description

Specifies whether and how the oscilloscope is configured to generate a CableSense signal on the specified channels when the [niScope_CableSenseSignalStart](#) function is called.

 Note The input impedance of the channel(s) to convey the CableSense signal must be set to 50 Ω.

Device-Specific Behavior

<p>PXIe-5160 PXIe-5162</p>	<ul style="list-style-type: none"> ▪ The value of this attribute must be identical across all channels whose input impedance is set to 50 Ω. ▪ If this attribute is set to a value other than <code>NISCOPE_VAL_CABLE_SENSE_DISABLED</code> for any channel(s), the input impedance of all channels for which this attribute is set to <code>NISCOPE_VAL_CABLE_SENSE_DISABLED</code> must be set to 1 MΩ.
--------------------------------	---

Supported Devices

- [PXIe-5110](#)
- [PXIe-5111](#)
- [PXIe-5113](#)
- [PXIe-5160](#)
- [PXIe-5162](#)

Defined Values

<code>NISCOPE_VAL_CABLE_SENSE_MODE_DISABLED</code>	(0) The oscilloscope is not configured to emit a CableSense signal.
<code>NISCOPE_VAL_CABLE_SENSE_MODE_ON_DEMAND</code>	(1) The oscilloscope is configured to emit a single CableSense pulse.
<code>NISCOPE_ATTR_CABLE_SENSE_VOLTAGE</code>	

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R	N/A	None	None

Description

Returns the voltage of the CableSense signal that is written to the EEPROM of the oscilloscope during factory calibration.

Supported Devices

- [PXle-5110](#)
- [PXle-5111](#)
- [PXle-5113](#)
- [PXle-5160](#)
- [PXle-5162](#)

NISCOPE_ATTR_BANDWIDTH

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	Channel	None	None

Description

Specifies the bandwidth of the channel. Express this value as the frequency (in hertz) at which the input circuitry attenuates the input signal by 3 dB.



Note This attribute is obsolete. Use [NISCOPE_ATTR_MAX_INPUT_FREQUENCY](#) for future programs.

NISCOPE_ATTR_CACHE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to cache the value of attributes. When caching is enabled, the instrument driver keeps track of the current instrument settings and avoids sending redundant commands to the instrument. Thus, you can significantly increase execution speed.

The instrument driver can choose to always cache or to never cache particular

attributes regardless of the setting of this attribute.

The default value is VI_TRUE. Use [niScope_InitWithOptions](#) to override the default.

Defined Values

VI_TRUE (1)

VI_FALSE (0)

NISCOPE_ATTR_CHANNEL_COUNT

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Indicates the number of channels that the specific instrument driver supports. For channel-based properties, the IVI engine maintains a separate cache value for each channel.

NISCOPE_ATTR_DRIVER_SETUP

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

This attribute indicates the Driver Setup string that you specified when initializing the driver.

Some cases exist where the end-user must specify instrument driver options at initialization. An example of this is specifying a particular instrument model from among a family of instruments that the driver supports. This is useful when using simulation. The end-user can specify driver-specific options through the

DriverSetup keyword in the optionsString parameter in `niScope_InitWithOptions`, or through the IVI Configuration Utility.

If you do not specify a Driver Setup string, this attribute returns an empty string.

NISCOPE_ATTR_GROUP_CAPABILITIES

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains a comma-separated list of class extension groups that NI-SCOPE implements.

NISCOPE_ATTR_INSTRUMENT_FIRMWARE_REVISION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the firmware revision information for the instrument.

NISCOPE_ATTR_INSTRUMENT_MANUFACTURER

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the name of the instrument manufacturer.

NISCOPE_ATTR_INSTRUMENT_MODEL

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the model number of the current instrument.

NISCOPE_ATTR_INTERCHANGE_CHECK

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

NI-SCOPE does not generate interchange warnings and therefore ignores this attribute.

NISCOPE_ATTR_LOGICAL_NAME

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string containing the logical name you specified when opening the current IVI session. You can pass a logical name to [niScope_Init](#) or [niScope_InitWithOptions](#). The IVI Configuration utility must contain an entry for the logical name. The logical

name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

NISCOPE_ATTR_MEAS_HIGH_REF

Data Type	Access	Channel-Based	Description
ViReal64	R/W	No	<p>An IVI-defined attribute that stores the high reference level used in many scalar measurements. You can use the function niScope_ConfigureRefLevels to write all the reference levels.</p> <p>This attribute is not channel based according to the IVI definition, so NI-SCOPE implements a channel-based version called NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL. Writing NISCOPE_ATTR_MEAS_HIGH_REF actually writes the same value to all channels of the channel based versions.</p> <p>Default Value 90%</p>

NISCOPE_ATTR_MEAS_LOW_REF

Data Type	Access	Channel-Based	Description
ViReal64	R/W	No	<p>This is an IVI-defined attribute that stores the low reference level used in many scalar measurements. You can use the function niScope_Confi</p>

[ConfigureRefLevels](#) to write all the reference levels. This attribute is not channel based according to the IVI definition, so NI-SCOPE implements a channel-based version called [NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL](#). Writing [NISCOPE_ATTR_MEAS_CHAN_LOW_REF](#) actually writes the same value to all channels of the channel based versions.

Default Value
10%

NISCOPE_ATTR_MEAS_MID_REF

Data Type	Access	Channel-Based	Description
ViReal64	R/W	No	<p>This is an IVI-defined attribute that stores the mid reference level used in many scalar measurement. You can use the function niScope_ConfigureRefLevels to write all the reference levels.</p> <p>This attribute is not channel based according to the IVI definition, so NI-SCOPE implements a Channel-Based version called NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL. Writing NISCOPE_ATTR</p>

R_MEAS_MID_REF actually writes the same value to all channels of the channel based versions.
 Default Value
 50%

NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the method used to map percentage reference units to voltages for the reference.

Default Value: NISCOPE_VAL_MEAS_BASE_TOP

Defined Values

NISCOPE_VAL_MEAS_LOW_HIGH

NISCOPE_VAL_MEAS_MIN_MAX

NISCOPE_VAL_MEAS_BASE_TOP

NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	Channel	None	None

Description

Specifies the units for the waveform measurement reference levels. If you choose percentage, the measurement routine uses the NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD attribute to map the percentage values to voltages. Choosing voltage units allows you to set the voltage thresholds directly and avoids extra calculations.

Default Value: Percentage

Defined Values

NISCOPE_VAL_MEAS_VOLTAGE—The reference levels are given in units of volts.

NISCOPE_VAL_MEAS_PERCENTAGE—Percentage, where the measurements voltage low and voltage high represent 0% and 100%, respectively.

NISCOPE_ATTR_NUM_CHANNELS

Data Type	Access	Channel- Based	Description
ViInt32	R/O	No	Indicates the number of channels that the specific instrument driver supports. For each attribute for which the IVI_VAL_MULTI_CHANNEL flag attribute is set, the IVI engine maintains a separate cache value for each channel.

NISCOPE_ATTR_QUERY_INSTRUMENT_STATUS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether the instrument driver queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this attribute to `VI_FALSE` to disable status checking and maximize performance.

The instrument driver can choose to ignore status checking for particular attributes regardless of the setting of this attribute. The default value is `VI_TRUE`. Use [niScope_InitWithOptions](#) to override the default.

Defined Values

`VI_TRUE`

`VI_FALSE`

`NISCOPE_ATTR_RANGE_CHECK`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to validate attribute values and function parameters. If enabled, the instrument driver validates the parameters values that you pass to driver functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this attribute to `VI_FALSE` to disable range checking and maximize performance.

The default value is `VI_TRUE`. Use [niScope_InitWithOptions](#) to override the default.

Defined Values

`VI_TRUE`

`VI_FALSE`

NISCOPE_ATTR_RECORD_COERCIONS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether the IVI engine keeps a list of the value coercions it makes for ViInt32 and ViReal64 attributes. all Ivi_GetNextCoercionInfo to extract and delete the oldest coercion record from the list. Use [niScope_InitWithOptions](#) to override the default value.

Default Value: VI_FALSE

Defined Values

VI_TRUE

VI_FALSE

Related topics:

- [Coercions of Horizontal Parameters](#)

NISCOPE_ATTR_RESOURCE_DESCRIPTOR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

Indicates the resource descriptor the driver uses to identify the physical device. If you initialize the driver with a logical name, this attribute contains the resource descriptor that corresponds to the entry in the IVI Configuration utility. If you

initialize the instrument driver with the resource descriptor, this attribute contains that value. You can pass a logical name to [niScope_Init](#) or [niScope_InitWithOptions](#). The IVI Configuration utility must contain an entry for the logical name. The logical name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

NISCOPE_ATTR_SIMULATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to simulate instrument driver I/O operations. If simulation is enabled, instrument driver functions perform range checking and call `Ivi_GetAttribute` and `Ivi_SetAttribute` functions, but they do not perform instrument I/O. For output parameters that represent instrument data, the instrument driver functions return calculated values.

Use [niScope_InitWithOptions](#) to override this value.

Default Value: VI_FALSE

Defined Values

VI_TRUE (1)

VI_FALSE (0)

NISCOPE_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

The major version number of the class specification with which this driver is compliant.

NISCOPE_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINC

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

The minor version number of the class specification with which this driver is compliant.

NISCOPE_ATTR_SPECIFIC_DRIVER_DESCRIPTION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains a brief description of the specific driver

NISCOPE_ATTR_SPECIFIC_DRIVER_PREFIX

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the prefix for the instrument driver. The name of each user-callable function in this driver starts with this prefix.

NISCOPE_ATTR_SPECIFIC_DRIVER_REVISION

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains additional version information about this instrument driver.

NISCOPE_ATTR_SPECIFIC_DRIVER_VENDOR

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains the name of the vendor that supplies this driver.

NISCOPE_ATTR_SUPPORTED_INSTRUMENT_MODELS

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

Description

A string that contains a comma-separated list of the instrument model numbers supported by this driver.

NISCOPE_ATTR_MAX_RIS_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the maximum sample rate in RIS mode in Hz.

Related topics:

- [Sampling Methods](#)
- [Equivalent-Time Sampling and Random Interleaved Sampling](#)

NISCOPE_ATTR_MAX_REAL_TIME_SAMPLING_RATE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

Description

Returns the maximum real time sample rate in Hz.

NISCOPE_ATTR_ONBOARD_MEMORY_SIZE

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	None

Description

Returns the total combined amount of onboard memory (that is, memory size) for all channels in bytes.

`NISCOPE_ATTR_DEVICE_NUMBER`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViInt32	R	N/A	None	None

Description

Returns the device number of the IF digitizer associated with the current session.

`NISCOPE_ATTR_FETCH_INTERLEAVED_DATA`

Specific Attribute

Data type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	None

Description

Specifies whether to interleave I and Q data retrieved from the IF digitizer in a single array or to retrieve two separate arrays, one for I data and another for Q data.

Default value: `VI_FALSE`

<code>VI_TRUE</code>	Retrieves a single array of alternating I and Q values. The resulting array is twice the size of the actual record length.
<code>VI_FALSE</code>	Retrieves two arrays, one for I values and another for Q values.

NI-SCOPE .NET Class Library Reference

Expand this topic to view information about programming with the NI-SCOPE .NET class library.

NI-SCOPE .NET Class Library Help

June 13, 2019, 375594C-01

You can use the NI-SCOPE .NET Class Library to interactively program NI high-speed oscilloscopes and digitizers. For additional information, refer to the NI High-Speed Digitizers Help. Select Start»All Programs»National Instruments»NI-SCOPE»NI-SCOPE Documentation to access the NI High-Speed Digitizers Help and NI-SCOPE Readme.

For additional information on developing applications using NI drivers and the .NET Framework, refer to ni.com/mstudio or visit ni.com/info and enter the Info Code NI`dot`NET.

You can find example applications by selecting National Instruments»NI-SCOPE»NI-SCOPE Examples in the Start menu.

© 2018–2019 National Instruments. All rights reserved.

Related information

[NIScope Namespace](#)

[NI High-Speed Digitizers Help](#)

Basic Usage

You can use the NI-SCOPE .NET class library to interactively program NI high-speed oscilloscopes and digitizers. The NI-SCOPE .NET API is an Interchangeable Virtual Instrument (IVI)-compliant instrument driver that features a set of methods and properties that exercise the functionality of the NI-SCOPE hardware.

1. **Initialization**—Use a constructor to create a new instance of the `NIScope` class. You must create an instance of this class before you can call into the underlying NI-SCOPE driver.

```
VB.NET

Using scopeSession As New NIScope("resourceName", false, false)
    ' Initialize a session to NI-SCOPE instruments.
    ' Your code goes here.
End Using
```

```
C#

using (NIScope scopeSession = new NIScope("resourceName", false, false))
{
    // Initialize a session to NI-SCOPE instruments.
    // Your code goes here.
}
```

2. **Configuration**—Use the `NIScope` object to access various sub-objects and to configure the NI-SCOPE driver directly or indirectly. You can use a sub-object of the `ScopeChannelCollection` class to configure channel-specific properties.

```
VB.NET

Dim scopeSession As New NIScope(ResourceName, False, False)
scopeSession.Channels(ChannelName).Configure(VerticalRange,
VerticalOffset, ScopeVerticalCoupling.DC, 1.0, True)
scopeSession.Timing.ConfigureTiming(sampleRateMin, recordLengthMin,
refPosition, 1, True)
Dim recordLength As Long = scopeSession.Acquisition.RecordLength
scopeSession.Measurement.Initiate()
Dim byteWaveforms As AnalogWaveformCollection(Of Byte) =
```

```
scopeSession.Channels(ChannelName).Measurement.FetchByte(timeout,
recordLength, byteWaveforms, waveformInfo)
```

C#

```
NIScope scopeSession = new NIScope(ResourceName, false, false);
scopeSession.Channels[ChannelName].Configure(VerticalRange,
VerticalOffset, ScopeVerticalCoupling.DC, 1.0, true);
scopeSession.Timing.ConfigureTiming(sampleRateMin, recordLengthMin,
refPosition, 1, true);
long recordLength = scopeSession.Acquisition.RecordLength;
scopeSession.Measurement.Initiate();
AnalogWaveformCollection<byte> byteWaveforms =
scopeSession.Channels[ChannelName].Measurement.FetchByte(timeout,
recordLength, byteWaveforms, out waveformInfo);
```

3. **Initiate**—NI-SCOPE devices start acquiring waveforms, causing the NIScope session to enter the running state.

VB.NET

```
scopeSession.Measurement.Initiate()
```

C#

```
scopeSession.Measurement.Initiate();
```

4. **Running State**—In the Running state, you can take measurements, fetch buffered measurements, query the output state, or query device states.

VB.NET

```
Dim recordLength As Long = scopeSession.Acquisition.RecordLength
Dim sampleRate As Double = scopeSession.Acquisition.SampleRate
Dim byteWaveforms As AnalogWaveformCollection(Of Byte) =
```

```
scopeSession.Channels(ChannelName).Measurement.FetchByte(timeout,
recordLength, byteWaveforms, waveformInfo)
```

C#

```
long recordLength = scopeSession.Acquisition.RecordLength;
double sampleRate = scopeSession.Acquisition.SampleRate;
AnalogWaveformCollection<byte> byteWaveforms
= scopeSession.Channels[ChannelName].Measurement.FetchByte(timeout,
recordLength, byteWaveforms, out waveformInfo);
```

5. **Close**—Closes the session to the device(s) and frees un-managed resources that are held by the session. You can close the session with **Close** or **Dispose**.

VB.NET

```
scopeSession.Close()
or
scopeSession.Dispose()
or
Using scopeSession As New NIScope("resourceName", false, false)
    ' Your code goes here.
End Using
```

C#

```
scopeSession.Close();
or
scopeSession.Dispose();
or
using (NIScope scopeSession = new NIScope("resourceName", false, false))
{
    // Your code goes here.
}
```

Refer to the Programming Flow topic in the NI High-Speed Digitizers Help for information about the specific NI-SCOPE software states.

Related information

[NIScope Class](#)

[NI-SCOPE Programming Flow](#)

How Do I Pass a Predefined Value to an NI-SCOPE Property?

There are certain properties, such as trigger sources and trigger and event output terminals, that work with a set of predefined string values, but might also accept custom values. For example, the Source property specifies a `ScopeDigitalEdgeStartTriggerSource` object. You can configure Source using one of the predefined values, such as:

VB.NET

```
scopeSession.Trigger.StartTrigger.DigitalEdge.Source =  
ScopeDigitalEdgeStartTriggerSource.Pfi0
```

C#

```
scopeSession.Trigger.StartTrigger.DigitalEdge.Source =  
ScopeDigitalEdgeStartTriggerSource.Pfi0;
```

You can also create a custom value with the static `FromString` method.

VB.NET

```
scopeSession.Trigger.StartTrigger.DigitalEdge.Source =  
ScopeDigitalEdgeStartTriggerSource.FromString("CustomSource")
```

```
C#

scopeSession.Trigger.StartTrigger.DigitalEdge.Source =
ScopeDigitalEdgeStartTriggerSource.FromString("CustomSource");
```

You can also directly set the source as a string instead of using the `FromString` method.

```
VB.NET

scopeSession.Trigger.StartTrigger.DigitalEdge.Source = "CustomSource"
```

```
C#

scopeSession.Trigger.StartTrigger.DigitalEdge.Source = "CustomSource";
```

How Do I Get the Underlying String from Source or Output Terminal Value?

You can use the `ToString` method to retrieve the underlying source values. For example, the following code shows how to retrieve the source name:

```
VB.NET

Dim triggerSource As String = ScopeDigitalEdgeStartTriggerSource.Pfi0.ToString()
```

```
C#

string triggerSource = ScopeDigitalEdgeStartTriggerSource.Pfi0.ToString();
```

You can use the underlying string value to assign a source or output terminal to some other source or output terminal.

Related information

[ScopeDigitalEdgeStartTrigger.Source Property](#)

How Do I Manage Errors and Warnings in the NI-SCOPE .NET Class Library?

The underlying NI-SCOPE driver reports any errors or warnings as error codes. Negative return values indicate errors, whereas positive values indicate warnings. In case of warnings, a warning event is raised that you can subscribe to.

Errors

The NI-SCOPE .NET API converts the negative error codes into exceptions and throws these exceptions. All exceptions that the API throws are either .NET-defined or IVI-defined; none of them are custom exceptions. The exception message for driver errors has the driver error code appended at the end.

Warnings

Warnings are communicated as events. To receive warnings, you must subscribe to the `Warning` event as follows:

- `_Driver Warning`

```
VB.NET
```

```
AddHandler scopeSession.DriverOperation.Warning, New EventHandler(Of  
ScopeWarningEventArgs) (AddressOf DriverOperation_Warning)  
Private Sub DriverOperation_Warning(ByVal sender As Object, ByVal e As  
ScopeWarningEventArgs)  
    'Code to handle Warnings.  
End Sub
```

```
C#
```

```
scopeSession.DriverOperation.Warning += new  
EventHandler<ScopeWarningEventArgs>(DriverOperation_Warning);  
void DriverOperation_Warning(object sender, ScopeWarningEventArgs e)
```

```
{
    //Code to handle Warnings.
}
```



Note National Instruments recommends subscribing to the warning event immediately after creating the NIScope object, to avoid missing any warnings that might occur.

- **_InterchangeCheck Warning**—InterchangeCheck warnings are communicated as events. To receive warnings, you must subscribe to the InterchangeCheckWarning event as follows:

```
VB.NET

AddHandler scopeSession.DriverOperation.InterchangeCheckWarning,
AddressOf DriverOperation_InterchangeCheckWarning
Private Sub DriverOperation_InterchangeCheckWarning(ByVal sender As
Object, ByVal e As ScopeInterchangeCheckWarningEventArgs)
    'Code to handle InterchangeCheckWarnings.
End Sub
```

```
C#

scopeSession.DriverOperation.InterchangeCheckWarning += new
EventHandler<ScopeInterchangeCheckWarningEventArgs>(DriverOperation_Interc
hangeCheckWarning);
void DriverOperation_InterchangeCheckWarning(object sender,
ScopeInterchangeCheckWarningEventArgs e)
{
    //Code to handle InterchangeCheckWarnings.
}
```

- **_Coercion Warning**—CoercionWarnings are communicated as events. To receive warnings, your must subscribe to the Coercion event as follows:


```

VB.NET

AddHandler scopeSession.DriverOperation.Coercion, AddressOf
DriverOperation_CoercionWarning
Private Sub DriverOperation_CoercionWarning(ByVal sender As Object, ByVal
e As ScopeCoercionEventArgs)
    'Code to handle CoercionWarnings.
End Sub

```

```

C#

scopeSession.DriverOperation.Coercion += new
EventHandler<ScopeCoercionEventArgs>(DriverOperation_CoercionWarning);
void DriverOperation_CoercionWarning(object sender,
ScopeCoercionEventArgs e)
{
    //Code to handle CoercionWarnings.
}

```

Each warning EventArgs class contains a read-only property Text to get the description of that event. For example:

- **Ivi.Driver.CoercionEventArgs**

```

VB.NET

Private Sub DriverOperation_CoercionWarning(ByVal sender As Object, ByVal
e As Ivi.Driver.CoercionEventArgs)
    Dim description As string = e.Text
End Sub

```

```

C#

void DriverOperation_CoercionWarning(object sender,

```

```
Ivi.Driver.CoercionEventArgs e)
{
    string description = e.Text;
}
```

- **`_Ivi.Driver.InterchangeCheckWarningEventArgs`**

```
VB.NET

Private Sub DriverOperation_InterchangeCheckWarning(ByVal sender As
Object, ByVal e As Ivi.Driver.InterchangeCheckWarningEventArgs)
    Dim description As string = e.Text
End Sub
```

```
C#

void DriverOperation_InterchangeCheckWarning(object sender,
Ivi.Driver.InterchangeCheckWarningEventArgs e)
{
    string description = e.Text;
}
```

Related information

[ScopeDriverOperation.Warning Event](#)

[ScopeDriverOperation.Coercion Event](#)

Using Fetch and Read Methods

The Fetch and Read methods are present in the `ScopeChannelMeasurement` class. A session must be opened, configured, and initiated before you can use Fetch and Read methods. Read methods internally invoke the `Initiate` method.

The Fetch and Read methods can be categorized into synchronous and asynchronous methods.

Using Async Methods

Some input/output (I/O) operations in NI-SCOPE can take a long time relative to other client program operations. In such cases, the NI-SCOPE .NET class library provides asynchronous programming support to execute the I/O operation on a different thread. This permits the client program to perform other operations without waiting for the I/O operation to complete.

The asynchronous I/O API in NI-SCOPE class library uses the Asynchronous Programming Model (APM) to implement the asynchronous I/O. In the APM, an asynchronous operation is implemented as two methods: **Begin <Operation>** and **End <Operation>**, where <Operation> is the name of an I/O method that operates synchronously.

After calling the **Begin <Operation>** method, a client program can continue executing instructions on the calling thread while the specific driver performs an <Operation> on a different thread. For each call to the **Begin <Operation>** method, the client program calls the **End <Operation>** method to get the results of the operation.

The following code shows an example of asynchronous I/O:

```
VB.NET

Private scopeSession As NIScope
    Private Sub StartAcquisition()
        Dim callBack As New AsyncCallback(AddressOf ProcessRead)
        scopeSession = New NIScope("scope1", True, True)
        scopeSession.Measurement.AutoSetup()
        ' Start the asynchronous Read operation.
        Dim timeout As New PrecisionTimeSpan(10.0)
        Dim asyncResult As IAsyncResult =
scopeSession.Channels("0").Measurement.BeginRead(timeout, -1, Nothing,
callBack, Nothing)
        ' ...
    End Sub
```

```

Private Sub ProcessRead(asyncResult As IAsyncResult)
    Dim waveforms As AnalogWaveformCollection(Of Double)
    Try
        ' Get the results.
        waveforms = scopeSession.Channels("0").Measurement.EndRead(asyncResult)
    Catch ex As Exception
        ' Handle the exception from the asynchronous Read operation.
    End Try
    ' ...
    ' Process waveforms.
    ' ...
End Sub

```

C#

```

NIScope scopeSession;
void StartAcquisition()
{
    AsyncCallback callBack = new AsyncCallback(ProcessRead);
    scopeSession = new NIScope("scope1", true, true);
    scopeSession.Measurement.AutoSetup();
    // Start the asynchronous Read operation.
    PrecisionTimeSpan timeout = new PrecisionTimeSpan(10.0);
    IAsyncResult asyncResult =
scopeSession.Channels["0"].Measurement.BeginRead(timeout, -1, null, callBack,
null);
    // ...
}

void ProcessRead(IAsyncResult asyncResult)
{
    AnalogWaveformCollection<double> waveforms;
    try
    {

```

```

    // Get the results.
    waveforms = scopeSession.Channels["0"].Measurement.EndRead(asyncResult);
}
catch (Exception ex)
{
    // Handle the exception from the asynchronous Read operation.
}
// ...
// Process waveforms.
// ...
}

```

Using Memory Optimization

All the Fetch and Read methods in NI-SCOPE .NET class library have a waveform parameter that reuses the memory across multiple Fetch or Read method calls. The following code snippet demonstrates this:

```

VB.NET

Using scopeSession As New NIScope("scope1", True, True)
    ' ...
    ' Configure scope channel "0" properties.
    ' ...

    ' First read operation.
    ' Here the waveform parameter is null, so the FetchDouble method
    ' internally allocates memory for the return waveform object.
    Dim waveform As AnalogWaveformCollection(Of Double) =
scopeSession.Channels("0").Measurement.FetchDouble(New PrecisionTimeSpan(10),
1000, Nothing)

    ' Second read operation. Notice the waveform is passed as a parameter to
the FetchDouble method.
    ' This time the FetchDouble method reuses memory of waveform object for the
returned waveform if possible.

```

```
    waveform = scopeSession.Channels("0").Measurement.FetchDouble(New  
PrecisionTimeSpan(10), 1000, waveform)  
End Using
```

```
C#  
  
using (NIScope scopeSession = new NIScope("scope1", true, true))  
{  
    // ...  
    // Configure scope channel "0" properties.  
    // ...  
  
    // First read operation.  
    // Here the waveform parameter is null, so the FetchDouble method  
    // internally allocates memory for the return waveform object.  
    AnalogWaveformCollection<double> waveform =  
scopeSession.Channels["0"].Measurement.FetchDouble(new PrecisionTimeSpan(10),  
1000, null);  
  
    // Second read operation. Notice the waveform is passed as a parameter to  
the FetchDouble method.  
    // This time the FetchDouble method reuses memory of waveform object for  
the returned waveform if possible.  
    waveform = scopeSession.Channels["0"].Measurement.FetchDouble(new  
PrecisionTimeSpan(10), 1000, waveform);  
}
```

Related information

[ScopeChannelMeasurement Class](#)

[Asynchronous Programming Model \(APM\)](#)

Mapping the NI-SCOPE .NET API to the NI-SCOPE C API

The following table maps the NI-SCOPE .NET API members to the corresponding NI-SCOPE C API and IVI .NET API members. All .NET members are in the `NationalInstruments.ModularInstruments.NIScope` namespace.

IVI .NET API Member	.NET API Member	C API Member
	<u>NIScope Class</u>	
<code>IVI.Driver.Close</code>	<u>Close</u>	<u>niScope_close</u>
	<u>ExportSignal</u>	<u>niScope_ExportSignal</u>
	<u>IsDeviceReady</u>	<u>niScope_IsDeviceReady</u>
	<u>ProbeCompensationSignalStart</u>	<u>niScope_ProbeCompensationSignalStart</u>
	<u>ProbeCompensationSignalStop</u>	<u>niScope_ProbeCompensationSignalStop</u>
	<u>ScopeAcquisition Class</u>	
	<u>Backlog</u>	<u>NISCOPE_ATTR_BACKLOG</u>
	<u>BinarySampleWidth</u>	<u>NISCOPE_ATTR_BINARY_SAMPLE_WIDTH</u>
	<u>DataTransferBandwidthMax</u>	<u>NISCOPE_ATTR_DATA_TRANSFER_MAXIMUM_B</u>
	<u>DataTransferBlockSize</u>	<u>NISCOPE_ATTR_DATA_TRANSFER_BLOCK_SIZE</u>
	<u>DataTransferPacketSizePreferred</u>	<u>NISCOPE_ATTR_DATA_TRANSFER_PREFERRED</u>
	<u>DdcDataProcessingMode</u>	<u>NISCOPE_ATTR_DDC_DATA_PROCESSING_MO</u>
	<u>NumberOfPointsMin</u>	<u>NISCOPE_ATTR_HORZ_MIN_NUM_PTS</u>
	<u>NumberOfMeasurementSamplesToFetch</u>	<u>NISCOPE_ATTR_FETCH_MEAS_NUM_SAMPLES</u>
	<u>OverflowErrorReportingMode</u>	<u>NISCOPE_ATTR_OVERFLOW_ERROR_REPORTIN</u>
	<u>RecordLength</u>	<u>NISCOPE_ATTR_HORZ_RECORD_LENGTH</u>
	<u>RecordNumberToFetch</u>	<u>NISCOPE_ATTR_FETCH_RECORD_NUMBER</u>
	<u>Resolution</u>	<u>NISCOPE_ATTR_RESOLUTION</u>
	<u>SampleMode</u>	<u>NISCOPE_ATTR_SAMPLE_MODE</u>
	<u>SampleRate</u>	<u>NISCOPE_ATTR_HORZ_SAMPLE_RATE</u>
	<u>SampleRateMin</u>	<u>NISCOPE_ATTR_MIN_SAMPLE_RATE</u>
	<u>StartTime</u>	<u>NISCOPE_ATTR_ACQUISITION_START_TIME</u>
	<u>TimePerRecord</u>	<u>NISCOPE_ATTR_HORZ_TIME_PER_RECORD</u>

IVI .NET API Member	.NET API Member	C API Member
	Type	NISCOPE_ATTR_ACQUISITION_TYPE
	ScopeAdvanceTrigger Class	
	SendSoftwareEdgeTrigger	niScope_SendSoftwareTriggerEdge
	ScopeAdvanceTriggerExportedOutputTerminal Class	NISCOPE_ATTR_EXPORTED_ADVANCE_TRIGGE
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeAdvanceTriggerSource Class	NISCOPE_ATTR_ADV_TRIG_SRC
	Immediate	NISCOPE_VAL_IMMEDIATE
	PxiStar	NISCOPE_VAL_PXI_STAR
	SoftwareTriggerFunction	NISCOPE_VAL_SW_TRIG_FUNC
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	ScopeArmReferenceTrigger Class	
	SendSoftwareEdgeTrigger	niScope_SendSoftwareTriggerEdge

IVI .NET API Member	.NET API Member	C API Member
	<u>ScopeArmReferenceTriggerSource</u> Class	<u>NISCOPE_ATTR_ARM_REF_TRIG_SRC</u>
	<u>Immediate</u>	<u>NISCOPE_VAL_IMMEDIATE</u>
	<u>PxiStar</u>	<u>NISCOPE_VAL_PXI_STAR</u>
	<u>SoftwareTriggerFunction</u>	<u>NISCOPE_VAL_SW_TRIG_FUNC</u>
	<u>Rtsi0</u>	<u>NISCOPE_VAL_RTSI_0</u>
	<u>Rtsi1</u>	<u>NISCOPE_VAL_RTSI_1</u>
	<u>Rtsi2</u>	<u>NISCOPE_VAL_RTSI_2</u>
	<u>Rtsi3</u>	<u>NISCOPE_VAL_RTSI_3</u>
	<u>Rtsi4</u>	<u>NISCOPE_VAL_RTSI_4</u>
	<u>Rtsi5</u>	<u>NISCOPE_VAL_RTSI_5</u>
	<u>Rtsi6</u>	<u>NISCOPE_VAL_RTSI_6</u>
	<u>Pfi0</u>	<u>NISCOPE_VAL_PFI_0</u>
	<u>Pfi1</u>	<u>NISCOPE_VAL_PFI_1</u>
	<u>ScopeCalibrationUtility</u> Class	
	<u>DeviceTemperature</u>	<u>NISCOPE_ATTR_DEVICE_TEMPERATURE</u>
	<u>ScopeChannel</u> Class	
	<u>BandPassFilterEnabled</u>	<u>NISCOPE_ATTR_BANDPASS_FILTER_ENABLED</u>
	<u>Coupling</u>	<u>NISCOPE_ATTR_VERTICAL_COUPLING</u>
	<u>DigitalGain</u>	<u>NISCOPE_ATTR_DIGITAL_GAIN</u>
	<u>DigitalOffset</u>	<u>NISCOPE_ATTR_DIGITAL_OFFSET</u>
	<u>DitherEnabled</u>	<u>NISCOPE_ATTR_DITHER_ENABLED</u>
	<u>Enabled</u>	<u>NISCOPE_ATTR_CHANNEL_ENABLED</u>
	<u>EnableInterleavingOffsetCorrection</u>	<u>NISCOPE_ATTR_INTERLEAVING_OFFSET_CORR</u>
	<u>EnableTimeInterleavedSampling</u>	<u>NISCOPE_ATTR_ENABLE_TIME_INTERLEAVED</u>
	<u>FlexFirAntiAliasFilterType</u>	<u>NISCOPE_ATTR_FLEX_FIR_ANTIALIAS_FILTER</u>
	<u>HighPassFilterFrequency</u>	<u>NISCOPE_ATTR_HIGH_PASS_FILTER_FREQUEN</u>
	<u>InputFrequencyMax</u>	<u>NISCOPE_ATTR_MAX_INPUT_FREQUENCY</u>
	<u>InputImpedance</u>	<u>NISCOPE_ATTR_INPUT_IMPEDANCE</u>
	<u>Offset</u>	<u>NISCOPE_ATTR_VERTICAL_OFFSET</u>
	<u>ProbeAttenuation</u>	<u>NISCOPE_ATTR_PROBE_ATTENUATION</u>
	<u>Range</u>	<u>NISCOPE_ATTR_VERTICAL_RANGE</u>

IVI .NET API Member	.NET API Member	C API Member
	TerminalConfiguration	NISCOPE_ATTR_CHANNEL_TERMINAL_CONFIG
	Configure	niScope_ConfigureVertical
	ConfigureCharacteristics	niScope_ConfigureChanCharacteristics
	GetFrequencyResponse	niScope_GetFrequencyResponse
	GetNormalizationCoefficients	niScope_GetNormalizationCoefficients
	GetScalingCoefficients	niScope_GetScalingCoefficients
	ScopeChannelDdc Class	
	CenterFrequency	NISCOPE_ATTR_DDC_CENTER_FREQUENCY
	Enabled	NISCOPE_ATTR_DDC_ENABLED
	FrequencyTranslationEnabled	NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATI
	FrequencyTranslationPhaseI	NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATI
	FrequencyTranslationPhaseQ	NISCOPE_ATTR_DDC_FREQUENCY_TRANSLATI
	QSource	NISCOPE_ATTR_DDC_Q_SOURCE
	ScopeChannelMeasurement Class	
	ArrayGain	NISCOPE_ATTR_MEAS_ARRAY_GAIN
	ArrayOffset	NISCOPE_ATTR_MEAS_ARRAY_OFFSET
	HysteresisPercent	NISCOPE_ATTR_MEAS_HYSTERESIS_PERCENT
	InterpolationSamplingFactor	NISCOPE_ATTR_MEAS_INTERPOLATION_SAMP
	LastHistogramSize	NISCOPE_ATTR_MEAS_LAST_ACQ_HISTOGRAM
	OtherChannel	NISCOPE_ATTR_MEAS_OTHER_CHANNEL
	PolynomialInterpolationOrder	NISCOPE_ATTR_MEAS_POLYNOMIAL_INTERPO
	AddWaveformProcessing	niScope_AddWaveformProcessing
	BeginFetchArrayMeasurement	niScope_FetchArrayMeasurement
	BeginFetchByte	niScope_FetchBinary8
	BeginFetchComplexDouble	niScope_FetchComplex
	BeginFetchComplexInt16	niScope_FetchComplexBinary16
	BeginFetchDouble	niScope_Fetch
	BeginFetchInt16	niScope_FetchBinary16
	BeginFetchInt32	niScope_FetchBinary32
	BeginFetchScalarMeasurement	niScope_FetchMeasurement
	BeginFetchScalarMeasurementStatistics	niScope_FetchMeasurementStats

IVI .NET API Member	.NET API Member	C API Member
	<u>BeginRead</u>	<u>niScope_Read</u>
	<u>BeginReadScalarMeasurement</u>	<u>niScope_ReadMeasurement</u>
	<u>ClearWaveformMeasurements</u>	<u>niScope_ClearWaveformMeasurementStats</u>
	<u>ClearWaveformProcessing</u>	<u>niScope_ClearWaveformProcessing</u>
	<u>EndFetchArrayMeasurement</u>	<u>niScope_FetchArrayMeasurement</u>
	<u>EndFetchByte</u>	<u>niScope_FetchBinary8</u>
	<u>EndFetchComplexDouble</u>	<u>niScope_FetchComplex</u>
	<u>EndFetchComplexInt16</u>	<u>niScope_FetchComplexBinary16</u>
	<u>EndFetchDouble</u>	<u>niScope_Fetch</u>
	<u>EndFetchInt16</u>	<u>niScope_FetchBinary16</u>
	<u>EndFetchInt32</u>	<u>niScope_FetchBinary32</u>
	<u>EndFetchScalarMeasurement</u>	<u>niScope_FetchMeasurement</u>
	<u>EndFetchScalarMeasurementStatistics</u>	<u>niScope_FetchMeasurementStats</u>
	<u>EndRead</u>	<u>niScope_Read</u>
	<u>EndReadScalarMeasurement</u>	<u>niScope_ReadMeasurement</u>
	<u>FetchArrayMeasurement</u>	<u>niScope_FetchArrayMeasurement</u>
	<u>FetchByte</u>	<u>niScope_FetchBinary8</u>
	<u>FetchInt16</u>	<u>niScope_FetchBinary16</u>
	<u>FetchInt32</u>	<u>niScope_FetchBinary32</u>
	<u>FetchComplexDouble</u>	<u>niScope_FetchComplex</u>
	<u>FetchComplexInt16</u>	<u>niScope_FetchComplexBinary16</u>
	<u>FetchDouble</u>	<u>niScope_Fetch</u>
	<u>FetchScalarMeasurement</u>	<u>niScope_FetchMeasurement</u>
	<u>FetchScalarMeasurementStatistics</u>	<u>niScope_FetchMeasurementStats</u>
	<u>Read</u>	<u>niScope_Read</u>
	<u>ReadScalarMeasurement</u>	<u>niScope_ReadMeasurement</u>
	<u>ScopeChannelMeasurementFilter</u> Class	
	<u>CenterFrequency</u>	<u>NISCOPE_ATTR_MEAS_FILTER_CENTER_FREQ</u>
	<u>CutoffFrequency</u>	<u>NISCOPE_ATTR_MEAS_FILTER_CUTOFF_FREQ</u>
	<u>FirFilterWindow</u>	<u>NISCOPE_ATTR_MEAS_FIR_FILTER_WINDOW</u>
	<u>Order</u>	<u>NISCOPE_ATTR_MEAS_FILTER_ORDER</u>

IVI .NET API Member	.NET API Member	C API Member
	Ripple	NISCOPE_ATTR_MEAS_FILTER_RIPPLE
	Taps	NISCOPE_ATTR_MEAS_FILTER_TAPS
	TransientPercent	NISCOPE_ATTR_MEAS_FILTER_TRANSIENT_WA
	Type	NISCOPE_ATTR_MEAS_FILTER_TYPE
	Width	NISCOPE_ATTR_MEAS_FILTER_WIDTH
	ScopeChannelMeasurementReferenceLevel Class	
	High	NISCOPE_ATTR_MEAS_CHAN_HIGH_REF_LEVEL
	Low	NISCOPE_ATTR_MEAS_CHAN_LOW_REF_LEVEL
	Mid	NISCOPE_ATTR_MEAS_CHAN_MID_REF_LEVEL
	PercentageMethod	NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD
	Units	NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS
	ScopeChannelMeasurementTimeHistogram Class	
	HighTime	NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIG
	HighVolts	NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_HIG
	LowTime	NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOV
	LowVolts	NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_LOV
	Size	NISCOPE_ATTR_MEAS_TIME_HISTOGRAM_SIZ
	ScopeChannelMeasurementVoltageHistogram Class	
	HighVolts	NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM
	LowVolts	NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM
	Size	NISCOPE_ATTR_MEAS_VOLTAGE_HISTOGRAM
	ScopeClockSynchronizationPulseSource Class	NISCOPE_ATTR_CLOCK_SYNC_PULSE_SOURC
	NoSource	NISCOPE_VAL_NO_SOURCE
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4

IVI .NET API Member	.NET API Member	C API Member
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeDigitalEdgeAdvanceTrigger Class	
	Source	NISCOPE_ATTR_ADV_TRIG_SRC
	ScopeDigitalEdgeArmReferenceTrigger Class	
	Source	NISCOPE_ATTR_ARM_REF_TRIG_SRC
	ScopeDigitalEdgeStartTrigger Class	
	Source	NISCOPE_ATTR_ACQ_ARM_SOURCE
	ScopeDigitalEdgeStartTriggerSource Class	NISCOPE_ATTR_ACQ_ARM_SOURCE
	Immediate	NISCOPE_VAL_IMMEDIATE
	PxiStar	NISCOPE_VAL_PXI_STAR
	SoftwareTriggerFunction	NISCOPE_VAL_SW_TRIG_FUNC
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeDriverOperation Class	
IlviDriverOperation.Cache	Cache	NISCOPE_ATTR_CACHE
IlviDriverOperation.Coercion	Coercion	

IVI .NET API Member	.NET API Member	C API Member
IlviDriverOperation.DriverSetup	DriverSetup	NISCOPE_ATTR_DRIVER_SETUP
	InterchangeCheck	NISCOPE_ATTR_INTERCHANGE_CHECK
IlviDriverOperation.IOResourceDescriptor	IOResourceDescriptor	NISCOPE_ATTR_RESOURCE_DESCRIPTOR
IlviDriverOperation.LogicalName	LogicalName	NISCOPE_ATTR_LOGICAL_NAME
IlviDriverOperation.QueryInstrumentStatus	QueryInstrumentStatus	NISCOPE_ATTR_QUERY_INSTRUMENT_STATUS
IlviDriverOperation.RangeCheck	RangeCheck	NISCOPE_ATTR_RANGE_CHECK
IlviDriverOperation.Simulate	Simulate	NISCOPE_ATTR_SIMULATE
IlviDriverOperation.InvalidateAllAttributes	InvalidateAllAttributes	niScope_InvalidateAllAttributes
IlviDriverOperation.ResetInterchangeCheck	ResetInterchangeCheck	niScope_ResetInterchangeCheck
IlviDriverOperation.InterchangeCheckWarning	InterchangeCheckWarning	
IlviDriverOperation.Warning	Warning	
	ScopeDriverUtility Class	
IlviDriverUtility.Disable	Disable	niScope_Disable
	ExportAttributeConfigurationBuffer	niScope_ExportAttributeConfigurationBuffer

IVI .NET API Member	.NET API Member	C API Member
	ExportAttributeConfigurationFile	niScope_ExportAttributeConfigurationFile
	FpgaBitFilePath	NISCOPE_ATTR_FPGA_BITFILE_PATH
	ImportAttributeConfigurationBuffer	niScope_ImportAttributeConfigurationBuffer
	ImportAttributeConfigurationFile	niScope_ImportAttributeConfigurationFile
	QueryDriverRevision	niScope_revision_query
IlviDriverUtility.Reset	Reset	niScope_reset
	ResetDevice	niScope_ResetDevice
IlviDriverUtility.ResetWithDefaults	ResetWithDefaults	niScope_ResetWithDefaults
IlviDriverUtility.SelfTest	SelfTest	niScope_self_test
IlviDriverUtility.ErrorQuery		
	ScopeEdgeTrigger Class	
	Slope	NISCOPE_ATTR_TRIGGER_SLOPE
	Configure	niScope_ConfigureTriggerEdge
	ScopeChannelEqualizationFilter Class	
	Enabled	NISCOPE_ATTR_EQUALIZATION_FILTER_ENABLED
	GetCoefficients	niScope_GetEqualizationFilterCoefficients
	NumberOfCoefficients	NISCOPE_ATTR_EQUALIZATION_NUM_COEFFICIENTS
	ConfigureCoefficients	niScope_ConfigureEqualizationFilterCoefficients
	ScopeDriverIdentity Class	
IlviComponentIdentity.Description	Description	NISCOPE_ATTR_SPECIFIC_DRIVER_DESCRIPTION
IlviDriverIdentity.Identifier	Identifier	
IlviDriverIdentity.InstrumentF	InstrumentFirmwareRevision	NISCOPE_ATTR_INSTRUMENT_FIRMWARE_REVISION

IVI .NET API Member	.NET API Member	C API Member
firmwareRevision		
IlviDriverIdentity.IlviDriverIdentity.InstrumentManufacturer	InstrumentManufacturer	NISCOPE_ATTR_INSTRUMENT_MANUFACTURE
IlviDriverIdentity.InstrumentModel	InstrumentModel	NISCOPE_ATTR_INSTRUMENT_MODEL
IlviComponentIdentity.Revision	Revision	NISCOPE_ATTR_SPECIFIC_DRIVER_REVISION
IlviComponentIdentity.SerialNumber	SerialNumber	NISCOPE_ATTR_SERIAL_NUMBER
IlviDriverIdentity.SpecificationMajorVersion	SpecificationMajorVersion	NISCOPE_ATTR_SPECIFIC_DRIVER_CLASS SPE
IlviDriverIdentity.SpecificationMinorVersion	SpecificationMinorVersion	NISCOPE_ATTR_SPECIFIC_DRIVER_CLASS SPE
IlviComponentIdentity.Vendor	Vendor	NISCOPE_ATTR_SPECIFIC_DRIVER_VENDOR
IlviDriverIdentity.GetGroupCapabilities	GetGroupCapabilities	NISCOPE_ATTR_GROUP_CAPABILITIES
IlviDriverIdentity.GetSupportedInstrumentModels	GetSupportedInstrumentModels	NISCOPE_ATTR_SUPPORTED_INSTRUMENT_M
	ScopeExportedSignals Class	
	AdvanceTriggerOutputTerminal	NISCOPE_ATTR_EXPORTED_ADVANCE TRIGGE
	EndOfAcquisitionEventOutputTerminal	NISCOPE_ATTR_END_OF_ACQUISITION EVENT
	EndOfRecordEventOutputTerminal	NISCOPE_ATTR_END_OF_RECORD_EVENT OU

IVI .NET API Member	.NET API Member	C API Member
	FiveVoltOutputTerminal	NISCOPE_ATTR_5V_OUT_OUTPUT_TERMINAL
	ReadyForAdvanceEventOutputTerminal	NISCOPE_ATTR_READY_FOR_ADVANCE_EVENT
	ReadyForReferenceEventOutputTerminal	NISCOPE_ATTR_READY_FOR_REF_EVENT_OUT
	ReadyForStartEventOutputTerminal	NISCOPE_ATTR_READY_FOR_START_EVENT_O
	ReferenceClockOutputTerminal	NISCOPE_ATTR_OUTPUT_CLOCK_SOURCE
	ReferenceTriggerOutputTerminal	NISCOPE_ATTR_EXPORTED_REF_TRIGGER_OU
	SampleClockOutputTerminal	NISCOPE_ATTR_EXPORTED_SAMPLE_CLOCK_C
	StartTriggerOutputTerminal	NISCOPE_ATTR_EXPORTED_START_TRIGGER_C
	ScopeEndOfAcquisitionEventOutputTerminal Class	NISCOPE_ATTR_END_OF_ACQUISITION_EVENT
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTISI_0
	Rtsi1	NISCOPE_VAL_RTISI_1
	Rtsi2	NISCOPE_VAL_RTISI_2
	Rtsi3	NISCOPE_VAL_RTISI_3
	Rtsi4	NISCOPE_VAL_RTISI_4
	Rtsi5	NISCOPE_VAL_RTISI_5
	Rtsi6	NISCOPE_VAL_RTISI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeEndOfRecordEventOutputTerminal Class	NISCOPE_ATTR_END_OF_RECORD_EVENT_OU
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTISI_0
	Rtsi1	NISCOPE_VAL_RTISI_1
	Rtsi2	NISCOPE_VAL_RTISI_2
	Rtsi3	NISCOPE_VAL_RTISI_3
	Rtsi4	NISCOPE_VAL_RTISI_4
	Rtsi5	NISCOPE_VAL_RTISI_5
	Rtsi6	NISCOPE_VAL_RTISI_6

IVI .NET API Member	.NET API Member	C API Member
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeExternalCalibration Class	
	ExternalCalibrationDate	niScope_CalFetchDate
	ExternalCalibrationTemperature	niScope_CalFetchTemperature
	ScopeFiveVoltsOutOutputTerminal Class	NISCOPE_ATTR_5V_OUT_OUTPUT_TERMINAL
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeFractionalResample Class	
	Enabled	NISCOPE_ATTR_FRACTIONAL_RESAMPLE_ENA
	ScopeInputClockSource Class	NISCOPE_ATTR_INPUT_CLOCK_SOURCE
	Aux0ClkIn	NISCOPE_VAL_AUX_0_CLK_IN
	NoSource	NISCOPE_VAL_NO_SOURCE
	RtsiClock	NISCOPE_VAL_RTSI_CLOCK
	PxiClock	NISCOPE_VAL_PXI_CLOCK
	External	NISCOPE_VAL_PXI_EXTERNAL
	ClkIn	NISCOPE_VAL_CLK_IN
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeMeasurement Class	

IVI .NET API Member	.NET API Member	C API Member
	<u>FetchInterleavedData</u>	<u>NISCOPE_ATTR_FETCH_INTERLEAVED_DATA</u>
	<u>FetchInterleavedIQData</u>	<u>NISCOPE_ATTR_FETCH_INTERLEAVED_IQ_DATA</u>
	<u>FetchOffset</u>	<u>NISCOPE_ATTR_FETCH_OFFSET</u>
	<u>FetchRelativeTo</u>	<u>NISCOPE_ATTR_FETCH_RELATIVE_TO</u>
	<u>PointsDone</u>	<u>NISCOPE_ATTR_POINTS_DONE</u>
	<u>RecordsDone</u>	<u>NISCOPE_ATTR_RECORDS_DONE</u>
	<u>RisInAutoSetupAllowed</u>	<u>NISCOPE_ATTR_RIS_IN_AUTO_SETUP_ENABLED</u>
	<u>Abort</u>	<u>niScope_Abort</u>
	<u>AutoSetup</u>	<u>niScope_AutoSetup</u>
	<u>Commit</u>	<u>niScope_Commit</u>
	<u>Initiate</u>	<u>niScope_InitiateAcquisition</u>
	<u>Status</u>	<u>niScope_AcquisitionStatus</u>
	<u>ScopeOutputClockSource</u> Class	<u>NISCOPE_ATTR_OUTPUT_CLOCK_SOURCE</u>
	<u>Aux0ClkOut</u>	<u>NISCOPE_VAL_AUX_0_CLK_OUT</u>
	<u>NoSource</u>	<u>NISCOPE_VAL_NO_SOURCE</u>
	<u>RtsiClock</u>	<u>NISCOPE_VAL_RTSI_CLOCK</u>
	<u>ClkOut</u>	<u>NISCOPE_VAL_CLK_OUT</u>
	<u>Pfi0</u>	<u>NISCOPE_VAL_PFI_0</u>
	<u>Pfi1</u>	<u>NISCOPE_VAL_PFI_1</u>
	<u>Pfi2</u>	<u>NISCOPE_VAL_PFI_2</u>
	<u>ScopeReadyForAdvanceEventOutputTerminal</u> Class	<u>NISCOPE_ATTR_READY_FOR_ADVANCE_EVENT</u>
	<u>PxiStar</u>	<u>NISCOPE_VAL_PXI_STAR</u>
	<u>Rtsi0</u>	<u>NISCOPE_VAL_RTSI_0</u>
	<u>Rtsi1</u>	<u>NISCOPE_VAL_RTSI_1</u>
	<u>Rtsi2</u>	<u>NISCOPE_VAL_RTSI_2</u>
	<u>Rtsi3</u>	<u>NISCOPE_VAL_RTSI_3</u>
	<u>Rtsi4</u>	<u>NISCOPE_VAL_RTSI_4</u>
	<u>Rtsi5</u>	<u>NISCOPE_VAL_RTSI_5</u>
	<u>Rtsi6</u>	<u>NISCOPE_VAL_RTSI_6</u>
	<u>Pfi0</u>	<u>NISCOPE_VAL_PFI_0</u>

IVI .NET API Member	.NET API Member	C API Member
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeReadyForReferenceEventOutputTerminal Class	NISCOPE_ATTR_READY_FOR_REF_EVENT_OUT
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeReadyForStartEventOutputTerminal Class	NISCOPE_ATTR_READY_FOR_START_EVENT_O
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeReferenceLevel Class	
	High	NISCOPE_ATTR_MEAS_HIGH_REF
	Low	NISCOPE_ATTR_MEAS_LOW_REF

IVI .NET API Member	.NET API Member	C API Member
	Mid	NISCOPE_ATTR_MEAS_MID_REF
	ScopeReferenceTrigger Class	
	Delay	NISCOPE_ATTR_TRIGGER_DELAY_TIME
	DetectorLocation	NISCOPE_ATTR_REF_TRIGGER_DETECTOR_LO
	Hysteresis	NISCOPE_ATTR_TRIGGER_HYSTERESIS
	QuietTimeMin	NISCOPE_ATTR_REF_TRIGGER_MINIMUM_QUI
	ReferencePosition	NISCOPE_ATTR_HORZ_RECORD_REF_POSITIO
	SendSoftwareEdgeTrigger	niScope_SendSoftwareTriggerEdge
	ScopeReferenceTriggerExportedOutputTerminal Class	NISCOPE_ATTR_EXPORTED_REF_TRIGGER_OU
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTISI_0
	Rtsi1	NISCOPE_VAL_RTISI_1
	Rtsi2	NISCOPE_VAL_RTISI_2
	Rtsi3	NISCOPE_VAL_RTISI_3
	Rtsi4	NISCOPE_VAL_RTISI_4
	Rtsi5	NISCOPE_VAL_RTISI_5
	Rtsi6	NISCOPE_VAL_RTISI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeSampleClockExportedOutputTerminal Class	NISCOPE_ATTR_EXPORTED_SAMPLE_CLOCK_C
	NoSource	NISCOPE_VAL_NO_SOURCE
	PxiStar	NISCOPE_VAL_PXI_STAR
	ClkOut	NISCOPE_VAL_CLK_OUT
	Rtsi0	NISCOPE_VAL_RTISI_0
	Rtsi1	NISCOPE_VAL_RTISI_1
	Rtsi2	NISCOPE_VAL_RTISI_2
	Rtsi3	NISCOPE_VAL_RTISI_3
	Rtsi4	NISCOPE_VAL_RTISI_4

IVI .NET API Member	.NET API Member	C API Member
	Rtsi5	NISCOPE_VAL_RTISI_5
	Rtsi6	NISCOPE_VAL_RTISI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	ScopeSampleClockTimebaseSource Class	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC
	ClkIn	NISCOPE_VAL_CLK_IN
	NoSource	NISCOPE_VAL_NO_SOURCE
	PxiStar	NISCOPE_VAL_PXI_STAR
	PxieDStarA	NISCOPE_VAL_PXIE_DSTAR_A
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	ScopeSelfCalibration Class	
	SelfCalibrate	niScope_CalSelfCalibrate
	SelfCalibrationDate	niScope_CalFetchDate
	SelfCalibrationTemperature	niScope_CalFetchTemperature
	ScopeStartTrigger Class	
	SendSoftwareEdgeTrigger	niScope_SendSoftwareTriggerEdge
	ScopeStartTriggerExportedOutputTerminal Class	NISCOPE_ATTR_EXPORTED_START_TRIGGER_C
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTISI_0
	Rtsi1	NISCOPE_VAL_RTISI_1
	Rtsi2	NISCOPE_VAL_RTISI_2
	Rtsi3	NISCOPE_VAL_RTISI_3
	Rtsi4	NISCOPE_VAL_RTISI_4
	Rtsi5	NISCOPE_VAL_RTISI_5
	Rtsi6	NISCOPE_VAL_RTISI_6
	Pfi0	NISCOPE_VAL_PFI_0
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	ScopeTiming Class	

IVI .NET API Member	.NET API Member	C API Member
	AbsoluteSampleClockOffset	NISCOPE_ATTR_ABSOLUTE_SAMPLE_CLOCK_OFFSET
	ClockSynchronizationPulseSource	NISCOPE_ATTR_CLOCK_SYNC_PULSE_SOURCE
	EnforceRealtime	NISCOPE_ATTR_HORZ_ENFORCE_REALTIME
	MoreRecordsThanMemoryAllowed	NISCOPE_ATTR_ALLOW_MORE_RECORDS_THAN_MEMORY_ALLOWED
	NumberOfRecordsToAcquire	NISCOPE_ATTR_HORZ_NUM_RECORDS
	OnboardMemorySize	NISCOPE_ATTR_ONBOARD_MEMORY_SIZE
	OscillatorPhaseDac	NISCOPE_ATTR_OSCILLATOR_PHASE_DAC_VALUE
	PllLockStatus	NISCOPE_ATTR_PLL_LOCK_STATUS
	PollInterval	NISCOPE_ATTR_POLL_INTERVAL
	RealtimeSampleRateMax	NISCOPE_ATTR_MAX_REAL_TIME_SAMPLING_RATE
	ReferenceClockRate	NISCOPE_ATTR_REF_CLK_RATE
	ReferenceClockSource	NISCOPE_ATTR_INPUT_CLOCK_SOURCE
	RisAverages	NISCOPE_ATTR_RIS_NUM_AVERAGES
	RisMethod	NISCOPE_ATTR_RIS_METHOD
	RisRateMax	NISCOPE_ATTR_MAX_RIS_RATE
	SampleClockTimebaseDivisor	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_DIV
	SampleClockTimebaseRate	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_RATE
	SampleClockTimebaseSource	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC
	SampleClockTimebaseMultiplier	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_MULT
	AdjustSampleClockRelativeDelay	niScope_AdjustSampleClockRelativeDelay
	ConfigureClock	niScope_ConfigureClock
	ConfigureTiming	niScope_ConfigureHorizontalTiming
	ScopeTrigger Class	
	Coupling	NISCOPE_ATTR_TRIGGER_COUPLING
	EndOfRecordToAdvanceTriggerHoldoff	NISCOPE_ATTR_END_OF_RECORD_TO_ADVANCE_TRIGGER_HOLDOFF
	ExternalTriggerSourceImpedance	NISCOPE_ATTR_TRIGGER_IMPEDANCE
	Holdoff	NISCOPE_ATTR_TRIGGER_HOLDOFF
	IsAutoTriggered	NISCOPE_ATTR_TRIGGER_AUTO_TRIGGERED
	Level	NISCOPE_ATTR_TRIGGER_LEVEL
	Modifier	NISCOPE_ATTR_TRIGGER_MODIFIER

IVI .NET API Member	.NET API Member	C API Member
	Source	NISCOPE_ATTR_TRIGGER_SOURCE
	StartToReferenceTriggerHoldoff	NISCOPE_ATTR_START_TO_REF_TRIGGER_HO
	TdcEnable	NISCOPE_ATTR_REF_TRIG_TDC_ENABLE
	Type	NISCOPE_ATTR_TRIGGER_TYPE
	Configure	niScope_ConfigureTrigger
	ConfigureTriggerDigital	niScope_ConfigureTriggerDigital
	ConfigureTriggerHysteresis	niScope_ConfigureTriggerHysteresis
	ConfigureTriggerImmediate	niScope_ConfigureTriggerImmediate
	ConfigureTriggerSoftware	niScope_ConfigureTriggerSoftware
	ConfigureTriggerWindow	niScope_ConfigureTriggerWindow
	ScopeTriggerSource Class	NISCOPE_ATTR_TRIGGER_SOURCE
	Aux0Pfi0	NISCOPE_VAL_AUX_0_PFI_0
	Aux0Pfi1	NISCOPE_VAL_AUX_0_PFI_1
	Aux0Pfi2	NISCOPE_VAL_AUX_0_PFI_2
	Aux0Pfi3	NISCOPE_VAL_AUX_0_PFI_3
	Aux0Pfi4	NISCOPE_VAL_AUX_0_PFI_4
	Aux0Pfi5	NISCOPE_VAL_AUX_0_PFI_5
	Aux0Pfi6	NISCOPE_VAL_AUX_0_PFI_6
	Aux0Pfi7	NISCOPE_VAL_AUX_0_PFI_7
	Immediate	NISCOPE_VAL_IMMEDIATE
	External	NISCOPE_VAL_EXTERNAL
	SoftwareTriggerFunction	NISCOPE_VAL_SW_TRIG_FUNC
	PxiStar	NISCOPE_VAL_PXI_STAR
	Rtsi0	NISCOPE_VAL_RTSI_0
	Rtsi1	NISCOPE_VAL_RTSI_1
	Rtsi2	NISCOPE_VAL_RTSI_2
	Rtsi3	NISCOPE_VAL_RTSI_3
	Rtsi4	NISCOPE_VAL_RTSI_4
	Rtsi5	NISCOPE_VAL_RTSI_5
	Rtsi6	NISCOPE_VAL_RTSI_6
	Pfi0	NISCOPE_VAL_PFI_0

IVI .NET API Member	.NET API Member	C API Member
	Pfi1	NISCOPE_VAL_PFI_1
	Pfi2	NISCOPE_VAL_PFI_2
	Channel0	0
	Channel1	1
	Channel2	2
	Channel3	3
	Channel4	4
	Channel5	5
	Channel6	6
	Channel7	7
	ScopeTVTrigger Class	
	DCRestoreEnabled	NISCOPE_ATTR_ENABLE_DC_RESTORE
	LineNumber	NISCOPE_ATTR_TV_TRIGGER_LINE_NUMBER
	Polarity	NISCOPE_ATTR_TV_TRIGGER_POLARITY
	SignalFormat	NISCOPE_ATTR_TV_TRIGGER_SIGNAL_FORMAT
	TriggerEvent	NISCOPE_ATTR_TV_TRIGGER_EVENT
	Configure	niScope_ConfigureTriggerVideo
	ScopeWindowReferenceTrigger Class	
	HighLevel	NISCOPE_ATTR_TRIGGER_WINDOW_HIGH_LEVEL
	LowLevel	NISCOPE_ATTR_TRIGGER_WINDOW_LOW_LEVEL
	Mode	NISCOPE_ATTR_TRIGGER_WINDOW_MODE

Mapping the NI-SCOPE .NET API Enums and Enum Values to the NI-SCOPE C API Attributes/Functions and Values

The following table maps the NI-SCOPE .NET API enums and enum values to the corresponding NI-SCOPE C API attributes and values. All .NET members are in the `NationalInstruments.ModularInstruments.NIScope` namespace.

.NET API Enums and Enum Values	C API Attributes/Functions and Values
<u>ScopeAcquisitionStatus</u>	<u>niScope_AcquisitionStatus</u>
Complete	NISCOPE_VAL_ACQ_COMPLETE
InProgress	NISCOPE_VAL_ACQ_IN_PROGRESS
Unknown	NISCOPE_VAL_ACQ_STATUS_UNKNOWN
<u>ScopeAcquisitionType</u>	<u>NISCOPE_ATTR_ACQUISITION_TYPE</u>
Ddc	NISCOPE_VAL_DDC
FlexibleResolution	NISCOPE_VAL_FLEXRES
Normal	NISCOPE_VAL_NORMAL
<u>ScopeArrayMeasurementType</u>	<u>niScope_FetchArrayMeasurement</u>
NoMeasurement	NISCOPE_VAL_NO_MEASUREMENT
LastAcquisitionHistogram	NISCOPE_VAL_LAST_ACQ_HISTOGRAM
FftPhaseSpectrum	NISCOPE_VAL_FFT_PHASE_SPECTRUM
FftAmplitudeSpectrumVoltsRms	NISCOPE_VAL_FFT_AMP_SPECTRUM_VOLTS_RMS
MultipleAcquisitionVoltageHistogram	NISCOPE_VAL_MULTI_ACQ_VOLTAGE_HISTOGRAM
MultipleAcquisitionTimeHistogram	NISCOPE_VAL_MULTI_ACQ_TIME_HISTOGRAM
ArrayIntegral	NISCOPE_VAL_ARRAY_INTEGRAL
Derivative	NISCOPE_VAL_DERIVATIVE
Inverse	NISCOPE_VAL_INVERSE
HanningWindow	NISCOPE_VAL_HANNING_WINDOW
FlatTopWindow	NISCOPE_VAL_FLAT_TOP_WINDOW
PolynomialInterpolation	NISCOPE_VAL_POLYNOMIAL_INTERPOLATION
MultiplyChannels	NISCOPE_VAL_MULTIPLY_CHANNELS
AddChannels	NISCOPE_VAL_ADD_CHANNELS
SubtractChannels	NISCOPE_VAL_SUBTRACT_CHANNELS
DivideChannels	NISCOPE_VAL_DIVIDE_CHANNELS
MultipleAcquisitionAverage	NISCOPE_VAL_MULTI_ACQ_AVERAGE
ButterworthFilter	NISCOPE_VAL_BUTTERWORTH_FILTER
ChebyshevFilter	NISCOPE_VAL_CHEBYSHEV_FILTER
FftAmplitudeSpectrumDB	NISCOPE_VAL_FFT_AMP_SPECTRUM_DB
HammingWindow	NISCOPE_VAL_HAMMING_WINDOW
WindowedFirFilter	NISCOPE_VAL_WINDOWED_FIR_FILTER

.NET API Enums and Enum Values	C API Attributes/Functions and Values
BesselFilter	NISCOPE_VAL_BESSEL_FILTER
TriangleWindow	NISCOPE_VAL_TRIANGLE_WINDOW
BlackmanWindow	NISCOPE_VAL_BLACKMAN_WINDOW
ArrayOffset	NISCOPE_VAL_ARRAY_OFFSET
ArrayGain	NISCOPE_VAL_ARRAY_GAIN
<u>ScopeChannelTerminalConfiguration</u>	<u>NISCOPE_ATTR_CHANNEL_TERMINAL_CONFIGURATION</u>
Differential	NISCOPE_VAL_DIFFERENTIAL
SingleEnded	NISCOPE_VAL_SINGLE_ENDED
UnbalancedDifferential	NISCOPE_VAL_UNBALANCED_DIFFERENTIAL
<u>ScopeDdcDataProcessingMode</u>	<u>NISCOPE_ATTR_DDC_DATA_PROCESSING_MODE</u>
Real	NISCOPE_VAL_REAL
Complex	NISCOPE_VAL_COMPLEX
<u>ScopeFetchRelativeTo</u>	<u>NISCOPE_ATTR_FETCH_RELATIVE_TO</u>
Now	NISCOPE_VAL_NOW
Pretrigger	NISCOPE_VAL_PRETRIGGER
ReadPointer	NISCOPE_VAL_READ_POINTER
Start	NISCOPE_VAL_START
Trigger	NISCOPE_VAL_TRIGGER
<u>ScopeFlexFirAntiAliasFilterType</u>	<u>NISCOPE_ATTR_FLEX_FIR_ANTIALIAS_FILTER_TYPE</u>
Tap16Hanning	NISCOPE_VAL_16_TAP_HANNING
Tap48Hanning	NISCOPE_VAL_48_TAP_HANNING
Tap48Standard	NISCOPE_VAL_48_TAP_STANDARD
Tap8Hanning	NISCOPE_VAL_8_TAP_HANNING
<u>ScopeMeasurementFilterType</u>	<u>NISCOPE_ATTR_MEAS_FILTER_TYPE</u>
LowPass	NISCOPE_VAL_MEAS_LOWPASS
HighPass	NISCOPE_VAL_MEAS_HIGHPASS
BandPass	NISCOPE_VAL_MEAS_BANDPASS
BandStop	NISCOPE_VAL_MEAS_BANDSTOP
<u>ScopeMeasurementFirFilterWindow</u>	<u>NISCOPE_ATTR_MEAS_FIR_FILTER_WINDOW</u>
None	NISCOPE_VAL_NONE
HanningWindow	NISCOPE_VAL_HANNING_WINDOW

.NET API Enums and Enum Values	C API Attributes/Functions and Values
FlatTopWindow	NISCOPE_VAL_FLAT_TOP_WINDOW
HammingWindow	NISCOPE_VAL_HAMMING_WINDOW
TriangleWindow	NISCOPE_VAL_TRIANGLE_WINDOW
BlackmanWindow	NISCOPE_VAL_BLACKMAN_WINDOW
<u>ScopeMeasurementPercentageMethod</u>	<u>NISCOPE_ATTR_MEAS_PERCENTAGE_METHOD</u>
LowHigh	NISCOPE_VAL_MEAS_LOW_HIGH
MinMax	NISCOPE_VAL_MEAS_MIN_MAX
BaseTop	NISCOPE_VAL_MEAS_BASE_TOP
<u>ScopeMeasurementReferenceLevelUnits</u>	<u>NISCOPE_ATTR_MEAS_REF_LEVEL_UNITS</u>
Voltage	NISCOPE_VAL_MEAS_VOLTAGE
Percentage	NISCOPE_VAL_MEAS_PERCENTAGE
<u>ScopeOverflowErrorReportingMode</u>	<u>NISCOPE_ATTR_OVERFLOW_ERROR_REPORTING</u>
Error	NISCOPE_VAL_ERROR_REPORTING_ERROR
Warning	NISCOPE_VAL_ERROR_REPORTING_WARNING
Disabled	NISCOPE_VAL_ERROR_REPORTING_DISABLED
<u>ScopeReferenceTriggerDetectorLocation</u>	<u>NISCOPE_ATTR_REF_TRIGGER_DETECTOR_LOCATION</u>
AnalogDetectionCircuit	NISCOPE_VAL_ANALOG_DETECTION_CIRCUIT
DdcOutput	NISCOPE_VAL_DDC_OUTPUT
<u>ScopeRisMethod</u>	<u>NISCOPE_ATTR_RIS_METHOD</u>
ExactNumberOfAverages	NISCOPE_VAL_RIS_EXACT_NUM_AVERAGES
MinimumNumberOfAverages	NISCOPE_VAL_RIS_MIN_NUM_AVERAGES
Incomplete	NISCOPE_VAL_RIS_INCOMPLETE
LimitedBinWidth	NISCOPE_VAL_RIS_LIMITED_BIN_WIDTH
<u>ScopeSampleMode</u>	<u>NISCOPE_ATTR_SAMPLE_MODE</u>
EquivalentTime	NISCOPE_VAL_EQUIVALENT_TIME
Realtime	NISCOPE_VAL_REAL_TIME
<u>ScopeScalarMeasurementType</u>	<u>niScope_FetchMeasurement</u>
RiseTime	NISCOPE_VAL_RISE_TIME
FallTime	NISCOPE_VAL_FALL_TIME
Frequency	NISCOPE_VAL_FREQUENCY
Period	NISCOPE_VAL_PERIOD

.NET API Enums and Enum Values	C API Attributes/Functions and Values
VoltageRMS	NISCOPE_VAL_VOLTAGE_RMS
VoltagePeakToPeak	NISCOPE_VAL_VOLTAGE_PEAK_TO_PEAK
VoltageMax	NISCOPE_VAL_VOLTAGE_MAX
VoltageMin	NISCOPE_VAL_VOLTAGE_MIN
VoltageHigh	NISCOPE_VAL_VOLTAGE_HIGH
VoltageLow	NISCOPE_VAL_VOLTAGE_LOW
VoltageAverage	NISCOPE_VAL_VOLTAGE_AVERAGE
WidthNegative	NISCOPE_VAL_WIDTH_NEG
WidthPositive	NISCOPE_VAL_WIDTH_POS
DutyCycleNegative	NISCOPE_VAL_DUTY_CYCLE_NEG
DutyCyclePositive	NISCOPE_VAL_DUTY_CYCLE_POS
VoltageAmplitude	NISCOPE_VAL_AMPLITUDE
VoltageCycleRms	NISCOPE_VAL_VOLTAGE_CYCLE_RMS
VoltageCycleAverage	NISCOPE_VAL_VOLTAGE_CYCLE_AVERAGE
Overshoot	NISCOPE_VAL_OVERSHOOT
Preshoot	NISCOPE_VAL_PRESHOOT
LowReferenceVolts	NISCOPE_VAL_LOW_REF_VOLTS
MidReferenceVolts	NISCOPE_VAL_MID_REF_VOLTS
HighReferenceVolts	NISCOPE_VAL_HIGH_REF_VOLTS
Area	NISCOPE_VAL_AREA
CycleArea	NISCOPE_VAL_CYCLE_AREA
Integral	NISCOPE_VAL_INTEGRAL
VoltageBase	NISCOPE_VAL_VOLTAGE_BASE
VoltageTop	NISCOPE_VAL_VOLTAGE_TOP
FftFrequency	NISCOPE_VAL_FFT_FREQUENCY
FftAmplitude	NISCOPE_VAL_FFT_AMPLITUDE
RiseSlewRate	NISCOPE_VAL_RISE_SLEW_RATE
FallSlewRate	NISCOPE_VAL_FALL_SLEW_RATE
ACEstimate	NISCOPE_VAL_AC_ESTIMATE
DCEstimate	NISCOPE_VAL_DC_ESTIMATE
TimeDelay	NISCOPE_VAL_TIME_DELAY

.NET API Enums and Enum Values	C API Attributes/Functions and Values
AveragePeriod	NISCOPE_VAL_AVERAGE_PERIOD
AverageFrequency	NISCOPE_VAL_AVERAGE_FREQUENCY
VoltageBaseToTop	NISCOPE_VAL_VOLTAGE_BASE_TO_TOP
PhaseDelay	NISCOPE_VAL_PHASE_DELAY
TimeHistogramMean	NISCOPE_VAL_TIME_HISTOGRAM_MEAN
TimeHistogramStandardDeviation	NISCOPE_VAL_TIME_HISTOGRAM_STDEV
TimeHistogramPeakToPeak	NISCOPE_VAL_TIME_HISTOGRAM_PEAK_TO_PEAK
TimeHistogramMedian	NISCOPE_VAL_TIME_HISTOGRAM_MEDIAN
TimeHistogramHits	NISCOPE_VAL_TIME_HISTOGRAM_HITS
TimeHistogramMax	NISCOPE_VAL_TIME_HISTOGRAM_MAX
TimeHistogramMin	NISCOPE_VAL_TIME_HISTOGRAM_MIN
TimeHistogramMeanPlusStandardDeviation	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_STDEV
TimeHistogramMeanPlus2StandardDeviation	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_2_STDEV
TimeHistogramMeanPlus3StandardDeviation	NISCOPE_VAL_TIME_HISTOGRAM_MEAN_PLUS_3_STDEV
TimeHistogramMode	NISCOPE_VAL_TIME_HISTOGRAM_MODE
TimeHistogramNewHits	NISCOPE_VAL_TIME_HISTOGRAM_NEW_HITS
VoltageHistogramMean	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN
VoltageHistogramStandardDeviation	NISCOPE_VAL_VOLTAGE_HISTOGRAM_STDEV
VoltageHistogramPeakToPeak	NISCOPE_VAL_VOLTAGE_HISTOGRAM_PEAK_TO_PEAK
VoltageHistogramMedian	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEDIAN
VoltageHistogramHits	NISCOPE_VAL_VOLTAGE_HISTOGRAM_HITS
VoltageHistogramMax	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MAX
VoltageHistogramMin	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MIN
VoltageHistogramMeanPlusStandardDeviation	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_STDEV
VoltageHistogramMeanPlus2StandardDeviation	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_2_STDEV
VoltageHistogramMeanPlus3StandardDeviation	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MEAN_PLUS_3_STDEV
VoltageHistogramMode	NISCOPE_VAL_VOLTAGE_HISTOGRAM_MODE













.NET API Enums and Enum Values	C API Attributes/Functions and Values
VoltageHistogramNewHits	NISCOPE_VAL_VOLTAGE_HISTOGRAM_NEW_HITS
<u>ScopeSelfCalibrationOption</u>	<u>niScope_CalSelfCalibrate</u>
NormalSelfCalibration	NISCOPE_VAL_SELF_CALIBRATION
RestorePreviousExternalCalibration	NISCOPE_VAL_CAL_RESTORE_EXTERNAL_CALIBRATION
<u>ScopeTriggerCoupling</u>	<u>NISCOPE_ATTR_TRIGGER_COUPLING</u>
AC	NISCOPE_VAL_AC
DC	NISCOPE_VAL_DC
HighFrequencyReject	NISCOPE_VAL_HF_REJECT
LowFrequencyReject	NISCOPE_VAL_LF_REJECT
ACPlusHighFrequencyReject	NISCOPE_VAL_AC_PLUS_HF_REJECT
<u>ScopeTriggerSlope</u>	<u>NISCOPE_ATTR_TRIGGER_SLOPE</u>
Positive	NISCOPE_VAL_POSITIVE
Negative	NISCOPE_VAL_NEGATIVE
<u>ScopeTriggerModifier</u>	<u>NISCOPE_ATTR_TRIGGER_MODIFIER</u>
None	NISCOPE_VAL_NO_TRIGGER_MOD
Auto	NISCOPE_VAL_AUTO
<u>ScopeTriggerType</u>	<u>NISCOPE_ATTR_TRIGGER_TYPE</u>
Edge	NISCOPE_VAL_EDGE_TRIGGER
Hysteresis	NISCOPE_VAL_HYSTERESIS_TRIGGER
DigitalEdge	NISCOPE_VAL_DIGITAL_TRIGGER
Window	NISCOPE_WINDOW_TRIGGER
Software	NISCOPE_VAL_SOFTWARE_TRIGGER
TV	NISCOPE_VAL_TV_TRIGGER
Immediate	NISCOPE_VAL_IMMEDIATE_TRIGGER
<u>ScopeTVTriggerEvent</u>	<u>NISCOPE_ATTR_TV_TRIGGER_EVENT</u>
Field1	NISCOPE_VAL_TV_EVENT_FIELD1
Field2	NISCOPE_VAL_TV_EVENT_FIELD2
AnyField	NISCOPE_VAL_TV_EVENT_ANY_FIELD
AnyLine	NISCOPE_VAL_TV_EVENT_ANY_LINE
LineNumber	NISCOPE_VAL_TV_EVENT_LINE_NUMBER
<u>ScopeTVTriggerPolarity</u>	<u>NISCOPE_ATTR_TV_TRIGGER_POLARITY</u>









.NET API Enums and Enum Values	C API Attributes/Functions and Values
Positive	NISCOPE_VAL_TV_POSITIVE
Negative	NISCOPE_VAL_TV_NEGATIVE
<u>ScopeTVTriggerSignalFormat</u>	<u>NISCOPE_ATTR_TV_TRIGGER_SIGNAL_FORMAT</u>
Ntsc	NISCOPE_VAL_NTSC
Pal	NISCOPE_VAL_Pal
Secam	NISCOPE_VAL_Secam
MPal	NISCOPE_VAL_M_PAL
I480Lines5994FieldsPerSecond	NISCOPE_VAL_480I_59_94_FIELDS_PER_SECOND
I480Lines60FieldsPerSecond	NISCOPE_VAL_480I_60_FIELDS_PER_SECOND
P480Lines5994FieldsPerSecond	NISCOPE_VAL_480P_59_94_FIELDS_PER_SECOND
P480Lines60FieldsPerSecond	NISCOPE_VAL_480P_60_FIELDS_PER_SECOND
I576Lines50FieldsPerSecond	NISCOPE_VAL_576I_50_FIELDS_PER_SECOND
P576Lines50FieldsPerSecond	NISCOPE_VAL_576P_50_FIELDS_PER_SECOND
P720Lines50FramesPerSecond	NISCOPE_VAL_720P_50_FRAMES_PER_SECOND
P720Lines5994FramesPerSecond	NISCOPE_VAL_720P_59_94_FRAMES_PER_SECOND
P720Lines60FramesPerSecond	NISCOPE_VAL_720P_60_FRAMES_PER_SECOND
I1080Lines50FieldsPerSecond	NISCOPE_VAL_1080I_50_FIELDS_PER_SECOND
I1080Lines5994FieldsPerSecond	NISCOPE_VAL_1080I_59_94_FIELDS_PER_SECOND
I1080Lines60FieldsPerSecond	NISCOPE_VAL_1080I_60_FIELDS_PER_SECOND
P1080Lines24FramesPerSecond	NISCOPE_VAL_1080P_24_FRAMES_PER_SECOND
<u>ScopeVerticalCoupling</u>	<u>NISCOPE_ATTR_VERTICAL_COUPLING</u>
AC	NISCOPE_VAL_AC
DC	NISCOPE_VAL_DC
Ground	NISCOPE_VAL_GND
<u>ScopeWindowTriggerMode</u>	<u>NISCOPE_ATTR_TRIGGER_WINDOW_MODE</u>
Entering	NISCOPE_VAL_ENTERING_WINDOW
Leaving	NISCOPE_VAL_LEAVING_WINDOW















NationalInstruments.ModularInstruments.NIScope Namespace

The NI-SCOPE .NET Class Library allows you to interactively program NI high-speed oscilloscopes and digitizers.










Classes

Class	Description
 NIScope	Defines a root class that is used to identify and control high-speed digitizers.
 ScopeAcquisition	Gets and sets the record to fetch from a channel that you specify.
 ScopeAdvanceTrigger	Represents the ScopeAdvanceTrigger sub-objects that contain properties and methods related to the advance trigger.
 ScopeAdvanceTriggerExportedOutputTerminal	Represents the output terminal for ScopeAdvanceTrigger .
 ScopeAdvanceTriggerSource	Represents the source terminal for ScopeDigitalEdgeAdvanceTrigger .
 ScopeArmReferenceTrigger	Represents the ScopeArmReferenceTrigger sub-objects that contain properties and methods related to the arm reference trigger.
 ScopeArmReferenceTriggerSource	Represents the source terminal for ScopeDigitalEdgeArmReferenceTrigger .
 ScopeCalibration	Represents the ScopeCalibration sub-objects that contain properties and methods related to calibration.
 ScopeCalibrationUtility	Represents the NI-SCOPE calibration utility.
 ScopeChannel	Represents a channel or a set of channels and the properties that you can set on a channel.
 ScopeChannelCollection	Represents a strongly-typed collection of ScopeChannelCollection objects.
 ScopeChannelDdc	Defines the properties related to the digital down converter (DDC) block in a high speed digitizer.
 ScopeChannelEqualizationFilter	Provides configuration options for the equalization filter. The equalization filter is designed to compensate the input signal for






		artifacts introduced to the signal outside of the digitizer.
 ScopeChannelMeasurement		Represents the ScopeChannelMeasurement sub-objects that contain properties to allow you to configure the digitizers to obtain scalar and array measurements.
 ScopeChannelMeasurementFilter		Represents the ScopeChannelMeasurement sub-objects that contain properties related to filtering.
 ScopeChannelMeasurementReferenceLevel		Represents the ScopeChannelMeasurement sub-objects that contain properties related to reference levels.
 ScopeChannelMeasurementTimeHistogram		Provides the properties related to time histograms, which place acquired samples into bins based on their time relative to a trigger point.
 ScopeChannelMeasurementVoltageHistogram		Represents the ScopeChannelMeasurement sub-objects that contain properties related to voltage histograms.
 ScopeChannelOnboardSignalProcessing		Provides methods and properties for configuring the Onboard Signal Processing (OSP) components of high speed digitizers.
 ScopeClockSynchronizationPulseSource		Represents the line on which the sample clock is sent or received.
 ScopeCoercionEventArgs		Represents the event data obtained when a coercion event occurs.
 ScopeDigitalEdgeAdvanceTrigger		Represents the configuration parameters for the digital-edge Advance trigger.
 ScopeDigitalEdgeArmReferenceTrigger		Represents the configuration parameters for the digital-edge arm Reference trigger, such as the trigger source and the amount of trigger delay.
 ScopeDigitalEdgeStartTrigger		Represents the ScopeDigitalEdgeStartTrigger sub-object that contains properties and methods related to the digital-edge Start trigger.
 ScopeDigitalEdgeStartTriggerSource		Represents the source terminal for ScopeDigitalEdgeStartTrigger .

 ScopeDriverIdentity	Provides the methods and properties that provide information about the instrument and the NI-SCOPE driver.
 ScopeDriverLock	Provides synchronization locks obtained on the driver session.
 ScopeDriverOperation	Provides properties that affect the operation of the instrument driver.
 ScopeDriverUtility	Contains methods that provide a basic set of utility operations.
 ScopeEdgeTrigger	Represents the sub-object that contains methods and properties used to configure common properties for analog edge triggering.
 ScopeEndOfAcquisitionEventOutputTerminal	Represents the output terminal for EndOfAcquisitionEventOutputTerminal .
 ScopeEndOfRecordEventOutputTerminal	Represents the end of record event output terminal for EndOfRecordEventOutputTerminal .
 ScopeExportedSignals	Represents the ScopeExportedSignals sub-objects that are used to configure the driver to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.
 ScopeExternalCalibration	Represents methods used to externally calibrate digitizer.
 ScopeFiveVoltsOutOutputTerminal	Represents the output terminal for FiveVoltOutputTerminal .
 ScopeFractionalResample	Provides configuration options for fractional resampling.
 ScopeGlitchTrigger	Represents the sub-object that contains methods and properties used to configure common properties for glitch triggering.
 ScopeInputClockSource	Represents the source terminal for ReferenceClockSource .
 ScopeInterchangeCheckWarningEventArgs	Provides NI-SCOPE interchange check warning event arguments.
















 ScopeMeasurement	Represents the NIScope sub-objects that contain the properties and methods related to acquisition.
 ScopeOutputClockSource	Represents the output terminal for ReferenceClockOutputTerminal .
 ScopeReadyForAdvanceEventOutputTerminal	Represents the output terminal for ReadyForAdvanceEventOutputTerminal .
 ScopeReadyForReferenceEventOutputTerminal	Represents the output terminal for ReadyForReferenceEventOutputTerminal .
 ScopeReadyForStartEventOutputTerminal	Represents the output terminal for ReadyForStartEventOutputTerminal .
 ScopeReferenceLevel	Provides properties and methods for configuring reference levels used in scalar measurements.
 ScopeReferenceTrigger	Represents the ScopeMeasurement sub-object that contains properties and methods that are related to reference triggers.
 ScopeReferenceTriggerExportedOutputTerminal	Represents the reference trigger output terminal for ReferenceTriggerOutputTerminal .
 ScopeRuntTrigger	Represents the sub-object that contains methods and properties used to configure common properties for runt triggering.
 ScopeSampleClockExportedOutputTerminal	Represents the output terminal for SampleClockOutputTerminal .
 ScopeSampleClockTimebaseSource	Represents the source terminal for SampleClockTimebaseSource .
 ScopeScalarMeasurementStatistics	Maintains the measurement value and the statistical results.
 ScopeSelfCalibration	Represents methods related to self calibration.
 ScopeStartTrigger	Represents the ScopeStartTrigger sub-objects that contain properties related to the Start trigger.
 ScopeStartTriggerExportedOutputTerminal	Represents the start trigger exported output terminal for StartTriggerOutputTerminal .











 ScopeSubObject	Represents the base class for all configuration sub-objects. This is an internal class and is not intended for external use.
 ScopeTiming	Represents the ScopeMeasurement sub-object that contains properties related to timing.
 ScopeTrigger	Represents the ScopeMeasurement sub-object that contains the properties and methods related to triggering in NI-SCOPE.
 ScopeTriggerSource	Represents the source terminal for Source .
 ScopeTVTrigger	Represents the ScopeMeasurement sub-object that contains properties and methods related to the video reference trigger.
 ScopeWarning	Provides warning codes for the warnings raised by the underlying driver.
 ScopeWarningEventArgs	Represents the properties under the Warning Event Args raised in case of a driver warning.
 ScopeWidthTrigger	Represents the sub-object that contains methods and properties used to configure common properties for width triggering.
 ScopeWindowReferenceTrigger	Represents sub-objects of ScopeWindowReferenceTrigger that contain properties related to the window reference trigger.

Structures

Structure	Description
 ScopeCoefficientInfo	Specifies an array of structures containing gain and offset coefficients for a given channel.
 ScopeErrorQueryResult	Represents the the result of an error query.
 ScopeRevisionQueryResult	Represents the result of the revision query operation.
 ScopeSelfTestResult	Represents the result of SelfTest .
 ScopeWaveformInfo	Contains timing and scaling information about the waveform obtained from the Fetch and Read methods.

Enumerations

Enumeration	Description
 ScopeAcquisitionStatus	Specifies the status information about the acquisition.
 ScopeAcquisitionType	Specifies how the digitizer acquires data and fills the waveform record.
 ScopeArrayMeasurementType	Specifies various array measurements.
 ScopeCableSenseMode	Specifies the CableSense signal type
 ScopeChannelTerminalConfiguration	Specifies the terminal configuration for the channel.
 ScopeDdcDataProcessingMode	Specifies how the digital downconverter (DDC) block of a high-speed digitizer should process the data.
 ScopeFetchRelativeTo	Specifies the position to start fetching within one record.
 ScopeFlexFirAntiAliasFilterType	Specifies the type of filter to achieve desired filtering characteristics. The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter.
 ScopeGlitchTriggerCondition	Specifies the polarity of the glitch that triggers the oscilloscope.
 ScopeMeasurementFilterType	Specifies the type of filter, for both IIR and FIR filters.
 ScopeMeasurementFirFilterWindow	Specifies the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.
 ScopeMeasurementPercentageMethod	Specifies the method used to map percentage reference units to voltages for the reference.
 ScopeMeasurementReferenceLevelUnits	Specifies the units of the waveform measurement reference levels. If you choose Percentage , then the measurement routine uses PercentageMethod to map the percentage values to voltages. By choosing Voltage , you can set the voltage threshold values directly and avoid extra calculations.
 ScopeOverflowErrorReportingMode	Controls the overflow error reporting behavior of the digital downconverter (DDC) block of the onboard signal processing (OSP) device.
 ScopeReferenceTriggerDetectorLocation	Specifies which reference trigger detection circuitry to use on a device that has an onboard signal processing (OSP) block.

 ScopeRisMethod	Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the maximum sampling rate.
 ScopeRuntTriggerCondition	Specifies whether a runt that is within or outside the high and low time limit triggers the oscilloscope.
 ScopeSampleMode	Specifies the sample mode the digitizer is currently using.
 ScopeScalarMeasurementType	Specifies various scalar measurements.
 ScopeSelfCalibrationOption	Specifies how the self-calibration option is carried out.
 ScopeSignalSource	Contains values representing signals to use with ExportSignal(ScopeSignalSource, String) .
 ScopeTriggerCoupling	Specifies how the digitizer couples the trigger source.
 ScopeTriggerModifier	Configures the device to automatically complete an acquisition if a trigger has not been received, after a period of time.
 ScopeTriggerPolarity	Specifies the polarity with which the digitizer will trigger.
 ScopeTriggerSlope	Specifies whether a rising or a falling edge triggers the digitizer.
 ScopeTriggerType	Specifies the type of trigger to use.
 ScopeTVTriggerEvent	Specifies the condition in the video signal that causes the digitizer to trigger.
 ScopeTVTriggerPolarity	Specifies whether a rising edge or a falling edge triggers the digitizer.
 ScopeTVTriggerSignalFormat	Specifies the type of video signal.
 ScopeVerticalCoupling	Specifies how the digitizer couples the input signal for the channel.
 ScopeWidthTriggerCondition	Specifies whether a pulse that is within or outside the high and low threshold triggers the oscilloscope.
 ScopeWindowTriggerMode	Specifies whether the trigger occurs when the signal enters or leaves a window.

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Class

Defines a root class that is used to identify and control high-speed digitizers.

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeNIScope

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




[C#VB](#)

```
public sealed class NIScope : ITClockSynchronizableDevice,
    IIviDriver, IServiceProvider, IDisposable, IIviScope
```

```
Public NotInheritable Class NIScope
    Implements ITClockSynchronizableDevice, IIviDriver, IServiceProvider,
    IDisposable,
    IIviScope
```













The NIScope type exposes the following members.

Constructors

	Name	Description
	NIScope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
	NIScope(String, Boolean, Boolean)	Initializes an instance of the NIScope class.
	NIScope(String, Boolean, Boolean, String)	Initializes an instance of the NIScope class.




[Top](#)












Properties

Name	Description
 Acquisition	Gets the ScopeAcquisition sub-object.
 Calibration	Gets the ScopeCalibration sub-object that contains properties related to calibration.
 Channels	Gets the collection of channels configured on the NI-SCOPE session.
 DriverOperation	Gets the ScopeDriverOperation sub-object that affects the operation of NI-SCOPE instrument driver.
 ExportedSignals	Gets the ScopeExportedSignals sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.
 Identity	Gets the ScopeDriverIdentity sub-object that provides identity and version information about the NI-SCOPE driver.
 IsDisposed	Gets a value that indicates whether the session has been disposed.
 Measurement	Gets the ScopeMeasurement sub-object.
 ReferenceLevel	Gets the ScopeReferenceLevel sub-object.
 Timing	Gets the ScopeTiming sub-object that contains properties that are related to timing.
 Trigger	Gets the ScopeTrigger sub-object that contains properties and sub-objects that are related to triggers.
 Utility	Gets the ScopeDriverUtility sub-object to access the utility features of NI-SCOPE.

[Top](#)

Methods

Name	Description
 CableSenseSignalStart	Enables the CableSense signal on any configured channels
 CableSenseSignalStop	Disables the CableSense signal on every channel
 Close	Performs the following actions after you finish using an instrument driver session: <ul style="list-style-type: none"> ▪ Closes the instrument I/O session. ▪ Destroys the IVI session and all of its attributes.

		<ul style="list-style-type: none"> Deallocates any memory resources used by the IVI session.
	Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	ExportSignal	Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetInstrumentHandle	Gets the SafeHandle to the NIScope instrument session.
	GetService	Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports AdvancedPropertyAccessService , IlviDriver , IlviScope , and NIScope .
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	IsDeviceReady	Determines whether the instrument is ready for use or the instrument is still undergoing initialization.
	ProbeCompensationSignalStart	Starts the 1 kHz square wave output on PFI 1 for probe compensation.
	ProbeCompensationSignalStop	Stops the 1 kHz square wave output on PFI 1 for probe compensation.
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

All properties, methods, and events fall under either the [NIScope](#) class or are the sub-objects of this class. To interact with NI-SCOPE, you must create an instance of this class.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor

Overload List

Name	Description
 NIScope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
 NIScope(String, Boolean, Boolean)	Initializes an instance of the NIScope class.
 NIScope(String, Boolean, Boolean, String)	Initializes an instance of the NIScope class.

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [SystemIntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

Remarks

resourceNameee accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

 Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean, String)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice,  
    string optionString  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean,  
    optionString As String  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

optionString

Type: [SystemString](#)

Sets the initial state of the following session properties: [RangeCheck](#), [QueryInstrumentStatus](#), [Cache](#), [Simulate](#). For more information on the format of optionString, refer to the [NI High-Speed Digitizers Help](#).

Remarks

resourceName accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)




[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor

Overload List

Name	Description
 NIScope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
 NIScope(String, Boolean, Boolean)	Initializes an instance of the NIScope class.
 NIScope(String, Boolean, Boolean, String)	Initializes an instance of the NIScope class.

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(
    IntPtr instrumentHandle
)
```

```
Public Sub New (
    instrumentHandle As IntPtr
)
```

Parameters

instrumentHandle

Type: [SystemIntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(
    IntPtr instrumentHandle
)
```

```
Public Sub New (
    instrumentHandle As IntPtr
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public NIScope(
    string resourceName,
    bool idQuery,
```

```
bool resetDevice
)
```

```
Public Sub New (
    resourceName As String,
    idQuery As Boolean,
    resetDevice As Boolean
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

Remarks

resourceNameee accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean, String)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice,  
    string optionString  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean,  
    optionString As String  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQueryType: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDeviceType: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

optionStringType: [SystemString](#)Sets the initial state of the following session properties: [RangeCheck](#), [QueryInstrumentStatus](#), [Cache](#), [Simulate](#). For more information on the format of optionString, refer to the [NI High-Speed Digitizers Help](#).**Remarks**

resourceName accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

 **Note**

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.













See Also**Reference**[NIScope Class](#)[NIScope Overload](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Properties

The [NIScope](#) type exposes the following members.

Properties

Name	Description
 Acquisition	Gets the ScopeAcquisition sub-object.
 Calibration	Gets the ScopeCalibration sub-object that contains properties related to calibration.
 Channels	Gets the collection of channels configured on the NI-SCOPE session.
 DriverOperation	Gets the ScopeDriverOperation sub-object that affects the operation of NI-SCOPE instrument driver.
 ExportedSignals	Gets the ScopeExportedSignals sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.
 Identity	Gets the ScopeDriverIdentity sub-object that provides identity and version information about the NI-SCOPE driver.
 IsDisposed	Gets a value that indicates whether the session has been disposed.
 Measurement	Gets the ScopeMeasurement sub-object.
 ReferenceLevel	Gets the ScopeReferenceLevel sub-object.
 Timing	Gets the ScopeTiming sub-object that contains properties that are related to timing.
 Trigger	Gets the ScopeTrigger sub-object that contains properties and sub-objects that are related to triggers.
 Utility	Gets the ScopeDriverUtility sub-object to access the utility features of NI-SCOPE.

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeAcquisition Property

Gets the [ScopeAcquisition](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAcquisition Acquisition { get; }
```

```
Public ReadOnly Property Acquisition As ScopeAcquisition  
    Get
```

Property Value

Type: [ScopeAcquisition](#)

Returns an object of the type [ScopeAcquisition](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeAcquisition Property

Gets the [ScopeAcquisition](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAcquisition Acquisition { get; }
```



```
Public ReadOnly Property Acquisition As ScopeAcquisition
    Get
```

Property Value

Type: [ScopeAcquisition](#)

Returns an object of the type [ScopeAcquisition](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeCalibration Property

Gets the [ScopeCalibration](#) sub-object that contains properties related to calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeCalibration Calibration { get; }
```

```
Public ReadOnly Property Calibration As ScopeCalibration
    Get
```

Property Value

Type: [ScopeCalibration](#)

Returns an object of the type [ScopeCalibration](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeChannels Property

Gets the collection of channels configured on the NI-SCOPE session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelCollection Channels { get; }
```

```
Public ReadOnly Property Channels As ScopeChannelCollection  
    Get
```

Property Value

Type: [ScopeChannelCollection](#)

Returns an object of type [ScopeChannelCollection](#).

Remarks

The Channels collection is initially empty. [ScopeChannel](#) instances are added to the collection when they are first indexed by channel name (e.g., "0" or "1"). While indexing channel groups by list (e.g., "0,1,2") or range (e.g., "0-2") produces [ScopeChannel](#) instances, the instances are not added to the collection and therefore cannot be enumerated using language features such as foreach.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeDriverOperation Property

Gets the [ScopeDriverOperation](#) sub-object that affects the operation of NI-SCOPE instrument driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverOperation DriverOperation { get; }
```

```
Public ReadOnly Property DriverOperation As ScopeDriverOperation  
    Get
```

Property Value

Type: [ScopeDriverOperation](#)

Returns an object of type [ScopeDriverOperation](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI ScopeExportedSignals Property

Gets the [ScopeExportedSignals](#) sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExportedSignals ExportedSignals { get; }
```

```
Public ReadOnly Property ExportedSignals As ScopeExportedSignals  
    Get
```

Property Value

Type: [ScopeExportedSignals](#)

Returns the [ScopeExportedSignals](#) sub-object that contains properties and sub-objects that are related to signals.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI ScopeIdentity Property

Gets the [ScopeDriverIdentity](#) sub-object that provides identity and version information about the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverIdentity Identity { get; }
```

```
Public ReadOnly Property Identity As ScopeDriverIdentity  
    Get
```

Property Value

Type: [ScopeDriverIdentity](#)

Returns an object of type [ScopeDriverIdentity](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeIsDisposed Property

Gets a value that indicates whether the session has been disposed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool IsDisposed { get; }
```

```
Public ReadOnly Property IsDisposed As Boolean
    Get
```

Property Value

Type: [Boolean](#)

if the session is disposed; otherwise, .

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeMeasurement Property

Gets the [ScopeMeasurement](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurement Measurement { get; }
```

```
Public ReadOnly Property Measurement As ScopeMeasurement
    Get
```

Property Value

Type: [ScopeMeasurement](#)

Returns an object of the type [ScopeMeasurement](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeReferenceLevel Property

Gets the [ScopeReferenceLevel](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceLevel ReferenceLevel { get; }
```

```
Public ReadOnly Property ReferenceLevel As ScopeReferenceLevel  
    Get
```

Property Value

Type: [ScopeReferenceLevel](#)

Returns an object of the type [ScopeReferenceLevel](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeTiming Property

Gets the [ScopeTiming](#) sub-object that contains properties that are related to timing.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTiming Timing { get; }
```

```
Public ReadOnly Property Timing As ScopeTiming  
    Get
```

Property Value

Type: [ScopeTiming](#)

Returns an object of the type [ScopeTiming](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeTrigger Property

Gets the [ScopeTrigger](#) sub-object that contains properties and sub-objects that are related to triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTrigger Trigger { get; }
```

```
Public ReadOnly Property Trigger As ScopeTrigger  
    Get
```

Property Value

Type: [ScopeTrigger](#)

Returns an object of the type [ScopeTrigger](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeUtility Property

Gets the [ScopeDriverUtility](#) sub-object to access the utility features of NI-SCOPE.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverUtility Utility { get; }
```

```
Public ReadOnly Property Utility As ScopeDriverUtility
    Get
```

Property Value

Type: [ScopeDriverUtility](#)

Returns an object of type [ScopeDriverUtility](#).

See Also

Reference

[NIScope Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)











Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Methods

The [NIScope](#) type exposes the following members.

Methods

	Name	Description
	CableSenseSignalStart	Enables the CableSense signal on any configured channels
	CableSenseSignalStop	Disables the CableSense signal on every channel
	Close	Performs the following actions after you finish using an instrument driver session: <ul style="list-style-type: none"> ▪ Closes the instrument I/O session. ▪ Destroys the IVI session and all of its attributes. ▪ Deallocates any memory resources used by the IVI session.
	Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 ExportSignal	Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetInstrumentHandle	Gets the SafeHandle to the NIScope instrument session.
 GetService	Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports AdvancedPropertyAccessService , IlviDriver , IlviScope , and NIScope .
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 IsDeviceReady	Determines whether the instrument is ready for use or the instrument is still undergoing initialization.
 ProbeCompensationSignalStart	Starts the 1 kHz square wave output on PFI 1 for probe compensation.
 ProbeCompensationSignalStop	Stops the 1 kHz square wave output on PFI 1 for probe compensation.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)[See Also](#)[Reference](#)[NIScope Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeCableSenseSignalStart Method

Enables the CableSense signal on any configured channels

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void CableSenseSignalStart()
```

```
Public Sub CableSenseSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeCableSenseSignalStart Method

Enables the CableSense signal on any configured channels

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void CableSenseSignalStart()
```

```
Public Sub CableSenseSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeCableSenseSignalStop Method

Disables the CableSense signal on every channel

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void CableSenseSignalStop()
```

```
Public Sub CableSenseSignalStop
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeClose Method

Performs the following actions after you finish using an instrument driver session:

- Closes the instrument I/O session.

- Destroys the IVI session and all of its attributes.
- Deallocates any memory resources used by the IVI session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Close()
```

```
Public Sub Close
```

Implements

IIviDriverClose

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeDispose Method

Closes the specified session and deallocates the reserved resources, if not already disposed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

IDisposable.Dispose

Remarks

You can call this method safely more than once, even if the session is already closed. A call to this method disposes the SafeHandle class used to hold the instrument handle. If the call to this method fails for some reason (e.g., the session was closed by some external means) you will not be notified about the failure. To help you identify failures in the ReleaseHandle method of the SafeHandle class, managed debugging assistant (MDA) is activated.

For details refer to <http://msdn.microsoft.com/en-us/library/85eak4a0.aspx>.

The call to this method fails when you externally close a session by:

- Initializing a session with a resource name for which the session is already open, within the same process. This causes the instrument handle held by the existing session to become invalid.
- Getting the instrument handle out using [GetInstrumentHandle](#) and closing this handle directly.

See Also

Reference

NI Scope Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

NIExportSignal Method

Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ExportSignal(
    ScopeSignalSource signalSource,
    string outputTerminal
)
```

```
Public Sub ExportSignal (
    signalSource As ScopeSignalSource,
    outputTerminal As String
)
```

Parameters

signalSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSignalSource](#)
The signal (clock, trigger, or event) to export.

outputTerminal

Type: [SystemString](#)
The hardware signal line on which the digital pulse is generated.

Exceptions

Exception	Condition
ObjectDisposedException	ExportSignal(ScopeSignalSource, String) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value of signalSource was invalid

- or -

The underlying NI-SCOPE driver returned an error.

Remarks

signalSource specifies what condition causes the digitizer to generate the signal. outputTerminal specifies where to send the signal on the hardware (such as a PFI connector or RTSI line). In cases where multiple instances of a particular signal exist, use the signal identifier input to specify which instance to control. You can call this method multiple times and set each line available to a different signal. To unprogram a specific line on an instrument, call this method with the signal you no longer want to export and set output terminal to [None](#).

Note

You may also use the properties on [ScopeExportedSignals](#) to export signals.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Triggering](#)

[SMC-Based Digitizers Acquisition Engine State Diagram](#)

[PXI Trigger Lines](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeGetInstrumentHandle Method

Gets the [SafeHandle](#) to the [NIScope](#) instrument session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public SafeHandle GetInstrumentHandle()
```

```
Public Function GetInstrumentHandle As SafeHandle
```

Return Value

Type: [SafeHandle](#)

The [SafeHandle](#) to the [NIScope](#) instrument session.

Exceptions

Exception	Condition
ObjectDisposedException	GetInstrumentHandle was called after the associated NIScope object was disposed.

Remarks

Using the [SafeHandle](#), you can get the [IntPtr](#) to the session; however, there are risks involved with using the [IntPtr](#). It is difficult to know the state of the handle, and the handle could be recycled while you are using it. For more information, refer to [DangerousGetHandle](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeGetService Method

Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports [AdvancedPropertyAccessService](#), [IlviDriver](#), [IlviScope](#), and [NIScope](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Object GetService(  
    Type serviceType  
)
```

```
Public Function GetService (  
    serviceType As Type  
) As Object
```

Parameters

serviceType

Type: [SystemType](#)

The type of the service object to get. It supports

[AdvancedPropertyAccessService](#), [IlviDriver](#), [IlviScope](#), and [NIScope](#).

Return Value

Type: [Object](#)

A service object of type serviceType, if one exists; otherwise, .

Implements

[IServiceProvider.GetService\(Type\)](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope IsDeviceReady Method

Determines whether the instrument is ready for use or the instrument is still undergoing initialization.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool IsDeviceReady(  
    string resourceName  
)
```

```
Public Shared Function IsDeviceReady (  
    resourceName As String  
) As Boolean
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument to be checked.

Return Value

Type: [Boolean](#)

if the instrument is ready to use; if the instrument is still initializing.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Probe Compensation Signal Start Method

Starts the 1 kHz square wave output on PFI 1 for probe compensation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ProbeCompensationSignalStart ()
```

```
Public Sub ProbeCompensationSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Probe Compensation Signal Stop Method

Stops the 1 kHz square wave output on PFI 1 for probe compensation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ProbeCompensationSignalStop()
```

```
Public Sub ProbeCompensationSignalStop
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Class

Defines a root class that is used to identify and control high-speed digitizers.

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeNIScope

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public sealed class NIScope : ITClockSynchronizableDevice,
    IIviDriver, IServiceProvider, IDisposable, IIviScope
```

```
Public NotInheritable Class NIScope
    Implements ITClockSynchronizableDevice, IIviDriver, IServiceProvider,
```




```

IDisposable,
    IIVI_Scope

```












The `NI_Scope` type exposes the following members.

Constructors

Name	Description
 NI_Scope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
 NI_Scope(String, Boolean, Boolean)	Initializes an instance of the <code>NI_Scope</code> class.
 NI_Scope(String, Boolean, Boolean, String)	Initializes an instance of the <code>NI_Scope</code> class.

[Top](#)











Properties



Name	Description
 Acquisition	Gets the ScopeAcquisition sub-object.
 Calibration	Gets the ScopeCalibration sub-object that contains properties related to calibration.
 Channels	Gets the collection of channels configured on the NI-SCOPE session.
 DriverOperation	Gets the ScopeDriverOperation sub-object that affects the operation of NI-SCOPE instrument driver.
 ExportedSignals	Gets the ScopeExportedSignals sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.
 Identity	Gets the ScopeDriverIdentity sub-object that provides identity and version information about the NI-SCOPE driver.
 IsDisposed	Gets a value that indicates whether the session has been disposed.
 Measurement	Gets the ScopeMeasurement sub-object.
 ReferenceLevel	Gets the ScopeReferenceLevel sub-object.
 Timing	Gets the ScopeTiming sub-object that contains properties that are related to timing.
 Trigger	Gets the ScopeTrigger sub-object that contains properties and sub-objects that are related to triggers.

	Utility	Gets the ScopeDriverUtility sub-object to access the utility features of NI-SCOPE.
---	-------------------------	--

[Top](#)

Methods

	Name	Description
	CableSenseSignalStart	Enables the CableSense signal on any configured channels
	CableSenseSignalStop	Disables the CableSense signal on every channel
	Close	<p>Performs the following actions after you finish using an instrument driver session:</p> <ul style="list-style-type: none"> ▪ Closes the instrument I/O session. ▪ Destroys the IVI session and all of its attributes. ▪ Deallocates any memory resources used by the IVI session.
	Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	ExportSignal	Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetInstrumentHandle	Gets the SafeHandle to the NIScope instrument session.
	GetService	Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports AdvancedPropertyAccessService , IlviDriver , IlviScope , and NIScope .
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	 IsDeviceReady	Determines whether the instrument is ready for use or the instrument is still undergoing initialization.
	ProbeCompensationSignalStart	Starts the 1 kHz square wave output on PFI 1 for probe compensation.

 ProbeCompensationSignalStop	Stops the 1 kHz square wave output on PFI 1 for probe compensation.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

Top

Remarks

All properties, methods, and events fall under either the `NIScope` class or are the sub-objects of this class. To interact with NI-SCOPE, you must create an instance of this class.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor

Overload List

Name	Description
 NIScope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
 NIScope(String, Boolean, Boolean)	Initializes an instance of the NIScope class.

 NIScope(String, Boolean, Boolean, String)	Initializes an instance of the NIScope class.
---	---

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

Remarks

resourceNameee accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean, String)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice,  
    string optionString  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean,  
    optionString As String  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

optionString

Type: [SystemString](#)

Sets the initial state of the following session properties: [RangeCheck](#), [QueryInstrumentStatus](#), [Cache](#), [Simulate](#). For more information on the format of optionString, refer to the [NI High-Speed Digitizers Help](#).

Remarks

resourceName accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor

Overload List

Name	Description
NIScope(IntPtr)	Creates a new instrument driver session from an existing instrument handle.
NIScope(String, Boolean, Boolean)	Initializes an instance of the NIScope class.
NIScope(String, Boolean, Boolean, String)	Initializes an instance of the NIScope class.

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (IntPtr)

Creates a new instrument driver session from an existing instrument handle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    IntPtr instrumentHandle  
)
```

```
Public Sub New (  
    instrumentHandle As IntPtr  
)
```

Parameters

instrumentHandle

Type: [System.IntPtr](#)

The pre-existing instrument handle used to create a new instrument driver session.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public NIScope(  
    string resourceName,  
    bool idQuery,  
    bool resetDevice  
)
```

```
Public Sub New (  
    resourceName As String,  
    idQuery As Boolean,  
    resetDevice As Boolean  
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

Remarks

resourceNameee accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Constructor (String, Boolean, Boolean, String)

Initializes an instance of the [NIScope](#) class.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public NIScope(
    string resourceName,
    bool idQuery,
    bool resetDevice,
    string optionString
)
```

```
Public Sub New (
    resourceName As String,
    idQuery As Boolean,
    resetDevice As Boolean,
    optionString As String
)
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument(s) to which the session is opened.

idQuery

Type: [SystemBoolean](#)

A value indicating whether you want the driver to perform an ID query. if the driver performs an ID query; otherwise .

resetDevice

Type: [SystemBoolean](#)

A value indicating whether you want to reset an instrument during the initialization procedure. if the instrument is reset; otherwise .

optionString

Type: [SystemString](#)

Sets the initial state of the following session properties: [RangeCheck](#), [QueryInstrumentStatus](#), [Cache](#), [Simulate](#). For more information on the format of optionString, refer to the [NI High-Speed Digitizers Help](#).

Remarks

resourceName accepts a comma-delimited list of strings in the form PXI1Slot3,PXI1Slot4, where PXI1Slot3 is one instrument resource name and PXI1Slot4 is another.

 Note

You can only specify multiple instruments of identical model numbers, bus types, channel counts, and onboard memory sizes. The instruments must be in the same chassis.

See Also

Reference

[NIScope Class](#)

[NIScope Overload](#)











[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

NIScope Properties

The [NIScope](#) type exposes the following members.

Properties

Name	Description
 Acquisition	Gets the ScopeAcquisition sub-object.
 Calibration	Gets the ScopeCalibration sub-object that contains properties related to calibration.
 Channels	Gets the collection of channels configured on the NI-SCOPE session.
 DriverOperation	Gets the ScopeDriverOperation sub-object that affects the operation of NI-SCOPE instrument driver.
 ExportedSignals	Gets the ScopeExportedSignals sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.
 Identity	Gets the ScopeDriverIdentity sub-object that provides identity and version information about the NI-SCOPE driver.
 IsDisposed	Gets a value that indicates whether the session has been disposed.
 Measurement	Gets the ScopeMeasurement sub-object.
 ReferenceLevel	Gets the ScopeReferenceLevel sub-object.
 Timing	Gets the ScopeTiming sub-object that contains properties that are related to timing.

 Trigger	Gets the ScopeTrigger sub-object that contains properties and sub-objects that are related to triggers.
 Utility	Gets the ScopeDriverUtility sub-object to access the utility features of NI-SCOPE.

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeAcquisition Property

Gets the [ScopeAcquisition](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public ScopeAcquisition Acquisition { get; }
```

```
Public ReadOnly Property Acquisition As ScopeAcquisition
    Get
```

Property Value

Type: [ScopeAcquisition](#)

Returns an object of the type [ScopeAcquisition](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeAcquisition Property

Gets the [ScopeAcquisition](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public ScopeAcquisition Acquisition { get; }
```

```
Public ReadOnly Property Acquisition As ScopeAcquisition  
    Get
```

Property Value

Type: [ScopeAcquisition](#)

Returns an object of the type [ScopeAcquisition](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Calibration Property

Gets the [ScopeCalibration](#) sub-object that contains properties related to calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCalibration Calibration { get; }
```

```
Public ReadOnly Property Calibration As ScopeCalibration  
    Get
```

Property Value

Type: [ScopeCalibration](#)

Returns an object of the type [ScopeCalibration](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Channels Property

Gets the collection of channels configured on the NI-SCOPE session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelCollection Channels { get; }
```

```
Public ReadOnly Property Channels As ScopeChannelCollection  
    Get
```

Property Value

Type: [ScopeChannelCollection](#)

Returns an object of type [ScopeChannelCollection](#).

Remarks

The Channels collection is initially empty. [ScopeChannel](#) instances are added to the collection when they are first indexed by channel name (e.g., "0" or "1"). While indexing channel groups by list (e.g., "0,1,2") or range (e.g., "0-2") produces [ScopeChannel](#) instances, the instances are not added to the collection and therefore cannot be enumerated using language features such as foreach.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeDriverOperation Property

Gets the [ScopeDriverOperation](#) sub-object that affects the operation of NI-SCOPE instrument driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverOperation DriverOperation { get; }
```

```
Public ReadOnly Property DriverOperation As ScopeDriverOperation  
    Get
```

Property Value

Type: [ScopeDriverOperation](#)

Returns an object of type [ScopeDriverOperation](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeExportedSignals Property

Gets the [ScopeExportedSignals](#) sub-object for configuring NI-SCOPE to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExportedSignals ExportedSignals { get; }
```

```
Public ReadOnly Property ExportedSignals As ScopeExportedSignals
    Get
```

Property Value

Type: [ScopeExportedSignals](#)

Returns the [ScopeExportedSignals](#) sub-object that contains properties and sub-objects that are related to signals.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeIdentity Property

Gets the [ScopeDriverIdentity](#) sub-object that provides identity and version information about the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverIdentity Identity { get; }
```

```
Public ReadOnly Property Identity As ScopeDriverIdentity
    Get
```

Property Value

Type: [ScopeDriverIdentity](#)

Returns an object of type [ScopeDriverIdentity](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeIsDisposed Property

Gets a value that indicates whether the session has been disposed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool IsDisposed { get; }
```

```
Public ReadOnly Property IsDisposed As Boolean  
    Get
```

Property Value

Type: [Boolean](#)

if the session is disposed; otherwise, .

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Measurement Property

Gets the [ScopeMeasurement](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurement Measurement { get; }
```

```
Public ReadOnly Property Measurement As ScopeMeasurement  
    Get
```

Property Value

Type: [ScopeMeasurement](#)

Returns an object of the type [ScopeMeasurement](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Reference Level Property

Gets the [ScopeReferenceLevel](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceLevel ReferenceLevel { get; }
```

```
Public ReadOnly Property ReferenceLevel As ScopeReferenceLevel
    Get
```

Property Value

Type: [ScopeReferenceLevel](#)

Returns an object of the type [ScopeReferenceLevel](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeTiming Property

Gets the [ScopeTiming](#) sub-object that contains properties that are related to timing.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTiming Timing { get; }
```

```
Public ReadOnly Property Timing As ScopeTiming
    Get
```

Property Value

Type: [ScopeTiming](#)

Returns an object of the type [ScopeTiming](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeTrigger Property

Gets the [ScopeTrigger](#) sub-object that contains properties and sub-objects that are related to triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTrigger Trigger { get; }
```

```
Public ReadOnly Property Trigger As ScopeTrigger  
    Get
```

Property Value

Type: [ScopeTrigger](#)

Returns an object of the type [ScopeTrigger](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeUtility Property

Gets the [ScopeDriverUtility](#) sub-object to access the utility features of NI-SCOPE.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public ScopeDriverUtility Utility { get; }
```

```
Public ReadOnly Property Utility As ScopeDriverUtility  
    Get
```

Property Value

Type: [ScopeDriverUtility](#)

Returns an object of type [ScopeDriverUtility](#).

See Also

Reference

[NIScope Class](#)














[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

NI Scope Methods

The [NIScope](#) type exposes the following members.

Methods

	Name	Description
	CableSenseSignalStart	Enables the CableSense signal on any configured channels
	CableSenseSignalStop	Disables the CableSense signal on every channel
	Close	Performs the following actions after you finish using an instrument driver session: <ul style="list-style-type: none"> ▪ Closes the instrument I/O session. ▪ Destroys the IVI session and all of its attributes. ▪ Deallocates any memory resources used by the IVI session.
	Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	ExportSignal	Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetInstrumentHandle	Gets the SafeHandle to the NIScope instrument session.
	GetService	Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports AdvancedPropertyAccessService , IlviDriver , IlviScope , and NIScope .
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	 IsDeviceReady	Determines whether the instrument is ready for use or the instrument is still undergoing initialization.
	ProbeCompensationSignalStart	Starts the 1 kHz square wave output on PFI 1 for probe compensation.

	ProbeCompensationSignalStop	Stops the 1 kHz square wave output on PFI 1 for probe compensation.
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeCableSenseSignalStart Method

Enables the CableSense signal on any configured channels

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public void CableSenseSignalStart()
```

```
Public Sub CableSenseSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI ScopeCableSenseSignalStart Method

Enables the CableSense signal on any configured channels

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void CableSenseSignalStart()
```

```
Public Sub CableSenseSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NI ScopeCableSenseSignalStop Method

Disables the CableSense signal on every channel

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void CableSenseSignalStop()
```

```
Public Sub CableSenseSignalStop
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeClose Method

Performs the following actions after you finish using an instrument driver session:

- Closes the instrument I/O session.
- Destroys the IVI session and all of its attributes.
- Deallocates any memory resources used by the IVI session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void Close()
```

```
Public Sub Close
```

Implements

[IIVIriverClose](#)

See Also

Reference

[NIScope Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Other Resources

Mapping the NI-SCOPE .NET API to the NI-SCOPE C API

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeDispose Method

Closes the specified session and deallocates the reserved resources, if not already disposed.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

IDisposable.Dispose

Remarks

You can call this method safely more than once, even if the session is already closed. A call to this method disposes the SafeHandle class used to hold the instrument handle. If the call to this method fails for some reason (e.g., the session was closed by some external means) you will not be notified about the failure. To help you identify failures in the ReleaseHandle method of the SafeHandle class, managed debugging assistant (MDA) is activated.

For details refer to <http://msdn.microsoft.com/en-us/library/85eak4a0.aspx>.

The call to this method fails when you externally close a session by:

- Initializing a session with a resource name for which the session is already open, within the same process. This causes the instrument handle held by the existing session to become invalid.
- Getting the instrument handle out using [GetInstrumentHandle](#) and closing this handle directly.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeExportSignal Method

Configures the digitizer to generate a signal that other instruments can detect when configured for digital triggering or sharing clocks.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ExportSignal(  
    ScopeSignalSource signalSource,  
    string outputTerminal  
)
```

```
Public Sub ExportSignal (  
    signalSource As ScopeSignalSource,  
    outputTerminal As String  
)
```

Parameters

signalSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSignalSource](#)

The signal (clock, trigger, or event) to export.

outputTerminal

Type: [SystemString](#)

The hardware signal line on which the digital pulse is generated.

Exceptions

Exception	Condition
ObjectDisposedException	ExportSignal(ScopeSignalSource, String) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value of signalSource was invalid - or - The underlying NI-SCOPE driver returned an error.

Remarks

signalSource specifies what condition causes the digitizer to generate the signal. outputTerminal specifies where to send the signal on the hardware (such as a PFI connector or RTSI line). In cases where multiple instances of a particular signal exist, use the signal identifier input to specify which instance to control. You can call this method multiple times and set each line available to a different signal. To unprogram a specific line on an instrument, call this method with the signal you no longer want to export and set output terminal to [None](#).

Note

You may also use the properties on [ScopeExportedSignals](#) to export signals.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Triggering](#)

[SMC-Based Digitizers Acquisition Engine State Diagram](#)

[PXI Trigger Lines](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeGetInstrumentHandle Method

Gets the [SafeHandle](#) to the [NIScope](#) instrument session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public SafeHandle GetInstrumentHandle()
```

```
Public Function GetInstrumentHandle As SafeHandle
```

Return Value

Type: [SafeHandle](#)

The [SafeHandle](#) to the [NIScope](#) instrument session.

Exceptions

Exception	Condition
ObjectDisposedException	GetInstrumentHandle was called after the associated NIScope object was disposed.

Remarks

Using the [SafeHandle](#), you can get the [IntPtr](#) to the session; however, there are risks involved with using the [IntPtr](#). It is difficult to know the state of the handle, and the

handle could be recycled while you are using it. For more information, refer to [DangerousGetHandle](#).

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope.GetService Method

Defines a mechanism for retrieving a service object (i.e., an object that provides custom support to other objects). It supports [AdvancedPropertyAccessService](#), [IlviDriver](#), [IlviScope](#), and [NIScope](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Object GetService(
    Type serviceType
)
```

```
Public Function GetService (
    serviceType As Type
) As Object
```

Parameters

serviceType

Type: [SystemType](#)

The type of the service object to get. It supports

[AdvancedPropertyAccessService](#), [IlviDriver](#), [IlviScope](#), and [NIScope](#).

Return Value

Type: [Object](#)

A service object of type `serviceType`, if one exists; otherwise, .

Implements

[IServiceProvider.GetService\(Type\)](#)

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScope.IsDeviceReady Method

Determines whether the instrument is ready for use or the instrument is still undergoing initialization.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool IsDeviceReady(  
    string resourceName  
)
```

```
Public Shared Function IsDeviceReady (  
    resourceName As String  
) As Boolean
```

Parameters

resourceName

Type: [SystemString](#)

The name of the instrument to be checked.

Return Value

Type: [Boolean](#)

if the instrument is ready to use; if the instrument is still initializing.

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeProbeCompensationSignalStart Method

Starts the 1 kHz square wave output on PFI 1 for probe compensation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ProbeCompensationSignalStart ()
```

```
Public Sub ProbeCompensationSignalStart
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

NIScopeProbeCompensationSignalStop Method

Stops the 1 kHz square wave output on PFI 1 for probe compensation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ProbeCompensationSignalStop()
```

```
Public Sub ProbeCompensationSignalStop
```

See Also

Reference

[NIScope Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisition Class

Gets and sets the record to fetch from a channel that you specify.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject
 NationalInstruments.ModularInstruments.NIScopeScopeAcquisition

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax






C#VB
















```
public sealed class ScopeAcquisition : ScopeSubObject,
    IIVI_ScopeAcquisition
```

```
Public NotInheritable Class ScopeAcquisition
    Inherits ScopeSubObject
    Implements IIVI_ScopeAcquisition
```

The ScopeAcquisition type exposes the following members.





Properties

Name	Description
 Backlog	Gets the number of (PointsDone) samples that have been acquired but not fetched for the record specified by RecordNumberToFetch .
 BinarySampleWidth	Gets or sets the bit width of the binary data in the acquired waveform, which is useful for determining which binary fetch method to use.
 DataTransferBandwidthMax	Gets or sets the maximum bandwidth that the device is allowed to consume.
 DataTransferBlockSize	Gets or sets the maximum number of samples to transfer at one time from the device to host memory.
 DataTransferPacketSizePreferred	Gets or sets the preferred size of the data field in the PCI Express (PCIe) packet during data transfer.

 DdcDataProcessingMode	Gets or sets the way in which data is processed by the digital downconverter (DDC).
 EnabledChannels	Returns a comma-separated list of the channels enabled for the session in ascending order.
 NumberOfMeasurementSamplesToFetch	Specifies the number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.
 NumberOfPointsMin	Gets or sets the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. RecordLength returns the actual record length.
 NumberOfRecordsToFetch	Gets or sets the number of records to fetch.
 OverflowErrorReportingMode	Gets or sets the reporting mode when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.
 RecordLength	Gets the actual number of points the digitizer acquires for each channel.
 RecordNumberToFetch	Gets or sets the record to fetch from a channel you specify.
 Resolution	Gets the bit width of valid data (as opposed to padding bits) in the acquired waveform.
 SampleMode	Gets the ScopeSampleMode the digitizer is currently using.
 SampleRate	Gets the effective sample rate using the current configuration in samples per second.
 SampleRateMin	Gets or sets the sampling rate for the acquisition in samples per second.
 StartTime	Specifies the length of time, in seconds, from the trigger event to the first point in the waveform record.
 TimePerRecord	Gets or sets the length of time that corresponds to the record length.
 Type	Specifies how the digitizer acquires data and fills the waveform record.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Acquisition Functions topic in the [NI High-Speed Digitizers Help](#) or to the Fetch and Read topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources













[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)









Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisition Properties

The [ScopeAcquisition](#) type exposes the following members.

Properties

Name	Description
 Backlog	Gets the number of (PointsDone) samples that have been acquired but not fetched for the record specified by RecordNumberToFetch .
 BinarySampleWidth	Gets or sets the bit width of the binary data in the acquired waveform, which is useful for determining which binary fetch method to use.
 DataTransferBandwidthMax	Gets or sets the maximum bandwidth that the device is allowed to consume.
 DataTransferBlockSize	Gets or sets the maximum number of samples to transfer at one time from the device to host memory.
 DataTransferPacketSizePreferred	Gets or sets the preferred size of the data field in the PCI Express (PCIe) packet during data transfer.
 DdcDataProcessingMode	Gets or sets the way in which data is processed by the digital downconverter (DDC).
 EnabledChannels	Returns a comma-separated list of the channels enabled for the session in ascending order.
 NumberOfMeasurementSamplesToFetch	Specifies the number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.
 NumberOfPointsMin	Gets or sets the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. RecordLength returns the actual record length.
 NumberOfRecordsToFetch	Gets or sets the number of records to fetch.
 OverflowErrorReportingMode	Gets or sets the reporting mode when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.
 RecordLength	Gets the actual number of points the digitizer acquires for each channel.

 RecordNumberToFetch	Gets or sets the record to fetch from a channel you specify.
 Resolution	Gets the bit width of valid data (as opposed to padding bits) in the acquired waveform.
 SampleMode	Gets the ScopeSampleMode the digitizer is currently using.
 SampleRate	Gets the effective sample rate using the current configuration in samples per second.
 SampleRateMin	Gets or sets the sampling rate for the acquisition in samples per second.
 StartTime	Specifies the length of time, in seconds, from the trigger event to the first point in the waveform record.
 TimePerRecord	Gets or sets the length of time that corresponds to the record length.
 Type	Specifies how the digitizer acquires data and fills the waveform record.

[Top](#)

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBacklog Property

Gets the number of ([PointsDone](#)) samples that have been acquired but not fetched for the record specified by [RecordNumberToFetch](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Backlog { get; }
```

```
Public ReadOnly Property Backlog As Double  
    Get
```

Property Value

Type: [Double](#)

The number of samples that have been acquired but not fetched for the record specified by record number to fetch.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBacklog Property

Gets the number of ([PointsDone](#)) samples that have been acquired but not fetched for the record specified by [RecordNumberToFetch](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Backlog { get; }
```

```
Public ReadOnly Property Backlog As Double
    Get
```

Property Value

Type: [Double](#)

The number of samples that have been acquired but not fetched for the record specified by record number to fetch.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBinarySampleWidth Property

Gets or sets the bit width of the binary data in the acquired waveform, which is useful for determining which binary fetch method to use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long BinarySampleWidth { get; set; }
```

```
Public Property BinarySampleWidth As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value of this property is device-dependent. Valid values are 8, 16, and 32.

Remarks

To configure the device to store samples with a lower resolution than the native resolution, set this property to the desired binary width. This process can be useful for streaming at faster speeds at the cost of the resolution. The least significant bits are lost with this configuration.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferBandwidthMax Property

Gets or sets the maximum bandwidth that the device is allowed to consume.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DataTransferBandwidthMax { get; set; }
```

```
Public Property DataTransferBandwidthMax As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The maximum bandwidth on the PCIe bus that is used during data transfer.

Remarks

The NI device limits itself to transfer fewer bytes per second on the PCIe bus than the value you specify for this property.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferBlockSize Property

Gets or sets the maximum number of samples to transfer at one time from the device to host memory.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public long DataTransferBlockSize { get; set; }
```

```
Public Property DataTransferBlockSize As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

Maximum number of samples to transfer at one time from the device to host memory.

Remarks

Increasing this number should result in better fetching performance because the driver does not need to restart the transfers as often. However, increasing this number may also increase the amount of page-locked memory required from the system.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferPacketSizePreferred

Gets or sets the preferred size of the data field in the PCI Express (PCIe) packet during data transfer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public int DataTransferPacketSizePreferred { get; set; }
```

```
Public Property DataTransferPacketSizePreferred As Integer
    Get
    Set
```

Property Value

Type: [Int32](#)

This property must be a power of two (64, 128,... , 512).

Remarks

In general, the larger the packet size, the more efficiently the device uses the bus. However, some systems, because of their implementation, perform better with smaller packet sizes.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDdcDataProcessingMode Property

Gets or sets the way in which data is processed by the digital downconverter (DDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeDdcDataProcessingMode DdcDataProcessingMode { get; set; }
```

```
Public Property DdcDataProcessingMode As ScopeDdcDataProcessingMode
    Get
    Set
```

Property Value

Type: [ScopeDdcDataProcessingMode](#)

Specifies the [ScopeDdcDataProcessingMode](#). The default value is [Complex](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with a device that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeDdcDataProcessingMode](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionEnabledChannels Property

Returns a comma-separated list of the channels enabled for the session in ascending order.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in

`NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public string EnabledChannels { get; }
```

```
Public ReadOnly Property EnabledChannels As String
    Get
```

Property Value

Type: [String](#)

A comma-separated list of the channels enabled for the session in ascending order.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfMeasurementSamplesToFetch

Specifies the number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfMeasurementSamplesToFetch { get; set; }
```

```
Public Property NumberOfMeasurementSamplesToFetch As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is -1.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfPointsMin Property

Gets or sets the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. [RecordLength](#) returns the actual record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfPointsMin { get; set; }
```

```
Public Property NumberOfPointsMin As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

Valid values range from 1 to the available onboard memory.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfRecordsToFetch Property

Gets or sets the number of records to fetch.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfRecordsToFetch { get; set; }
```

```
Public Property NumberOfRecordsToFetch As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is -1.

Remarks

If you want to fetch all records from the record you specify in [RecordNumberToFetch](#) to the last record configured, use -1.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionOverflowErrorReportingMode Property

Gets or sets the reporting mode when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeOverflowErrorReportingMode OverflowErrorReportingMode { get; set; }
```

```
Public Property OverflowErrorReportingMode As ScopeOverflowErrorReportingMode
    Get
    Set
```

Property Value

Type: [ScopeOverflowErrorReportingMode](#)

Specifies the [ScopeOverflowErrorReportingMode](#). The default value is [Warning](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI PXI-5142. If you open a session with a device that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionRecordLength Property

Gets the actual number of points the digitizer acquires for each channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordLength { get; }
```

```
Public ReadOnly Property RecordLength As Long  
    Get
```

Property Value

Type: [Int64](#)

The value is equal to or greater than the minimum number of points you specify with [NumberOfPointsMin](#).

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionRecordNumberToFetch Property

Gets or sets the record to fetch from a channel you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordNumberToFetch { get; set; }
```

```
Public Property RecordNumberToFetch As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 0. Valid values are greater than or equal to 0.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionResolution Property

Gets the bit width of valid data (as opposed to padding bits) in the acquired waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Resolution { get; }
```

```
Public ReadOnly Property Resolution As Long  
    Get
```

Property Value

Type: [Int64](#)

The bit width of valid data in the acquired waveform.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleMode Property

Gets the [ScopeSampleMode](#) the digitizer is currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleMode SampleMode { get; }
```

```
Public ReadOnly Property SampleMode As ScopeSampleMode  
    Get
```

Property Value

Type: [ScopeSampleMode](#)

Specifies the [ScopeSampleMode](#).

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleRate Property

Gets the effective sample rate using the current configuration in samples per second.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleRate { get; }
```

```
Public ReadOnly Property SampleRate As Double  
    Get
```

Property Value

Type: [Double](#)

The effective sample rate using the current configuration in samples per second.

Implements

[IIVI_ScopeAcquisitionSampleRate](#)

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleRateMin Property

Gets or sets the sampling rate for the acquisition in samples per second.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleRateMin { get; set; }
```

```
Public Property SampleRateMin As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The combination of this property and [NumberOfPointsMin](#) must allow the digitizer to sample at a valid sampling rate and not require more memory than the onboard memory module allows.

Remarks

This property is the frequency at which digitized samples are stored, expressed in samples per second. This property is rounded up to the next valid sampling rate that the device supports. If [EnforceRealtime](#) is set to `True`, the sampling rate must be set lower than the [RealtimeSampleRateMax](#) of the digitizer, so data can be digitized at the requested rate. If [EnforceRealtime](#) is set to `False`, sampling rates can be set higher than the maximum real-time sampling rate of the digitizer. When this happens, NI-

SCOPE enters random interleaved sampling (RIS) mode, where it acquires multiple waveforms at the maximum real-time sampling rate and reconstructs a periodic waveform.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionStartTime Property

Specifies the length of time, in seconds, from the trigger event to the first point in the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan StartTime { get; set; }
```

```
Public Property StartTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Length of time from the trigger event to the first point in the waveform record in seconds.

Remarks

If the value is positive, the first point in the waveform record occurs after the trigger event. If the value is negative, the first point in the waveform record occurs before the trigger event.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionTimePerRecord Property

Gets or sets the length of time that corresponds to the record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimePerRecord { get; set; }
```

```
Public Property TimePerRecord As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The length of time that corresponds to the record length.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionType Property

Specifies how the digitizer acquires data and fills the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAcquisitionType Type { get; set; }
```

```
Public Property Type As ScopeAcquisitionType  
    Get  
    Set
```

Property Value

Type: [ScopeAcquisitionType](#)

Specifies the [ScopeAcquisitionType](#).

See Also

Reference

[ScopeAcquisition Class](#)












[NationalInstruments.ModularInstruments.NIScope Namespace](#)










Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisition Properties

The [ScopeAcquisition](#) type exposes the following members.

Properties

Name	Description
 Backlog	Gets the number of (PointsDone) samples that have been acquired but not fetched for the record specified by RecordNumberToFetch .
 BinarySampleWidth	Gets or sets the bit width of the binary data in the acquired waveform, which is useful for determining which binary fetch method to use.
 DataTransferBandwidthMax	Gets or sets the maximum bandwidth that the device is allowed to consume.
 DataTransferBlockSize	Gets or sets the maximum number of samples to transfer at one time from the device to host memory.
 DataTransferPacketSizePreferred	Gets or sets the preferred size of the data field in the PCI Express (PCIe) packet during data transfer.
 DdcDataProcessingMode	Gets or sets the way in which data is processed by the digital downconverter (DDC).
 EnabledChannels	Returns a comma-separated list of the channels enabled for the session in ascending order.
 NumberOfMeasurementSamplesToFetch	Specifies the number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.
 NumberOfPointsMin	Gets or sets the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. RecordLength returns the actual record length.
 NumberOfRecordsToFetch	Gets or sets the number of records to fetch.
 OverflowErrorReportingMode	Gets or sets the reporting mode when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.

 RecordLength	Gets the actual number of points the digitizer acquires for each channel.
 RecordNumberToFetch	Gets or sets the record to fetch from a channel you specify.
 Resolution	Gets the bit width of valid data (as opposed to padding bits) in the acquired waveform.
 SampleMode	Gets the ScopeSampleMode the digitizer is currently using.
 SampleRate	Gets the effective sample rate using the current configuration in samples per second.
 SampleRateMin	Gets or sets the sampling rate for the acquisition in samples per second.
 StartTime	Specifies the length of time, in seconds, from the trigger event to the first point in the waveform record.
 TimePerRecord	Gets or sets the length of time that corresponds to the record length.
 Type	Specifies how the digitizer acquires data and fills the waveform record.

[Top](#)

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBacklog Property

Gets the number of ([PointsDone](#)) samples that have been acquired but not fetched for the record specified by [RecordNumberToFetch](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Backlog { get; }
```

```
Public ReadOnly Property Backlog As Double  
    Get
```

Property Value

Type: [Double](#)

The number of samples that have been acquired but not fetched for the record specified by record number to fetch.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBacklog Property

Gets the number of ([PointsDone](#)) samples that have been acquired but not fetched for the record specified by [RecordNumberToFetch](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public double Backlog { get; }
```

```
Public ReadOnly Property Backlog As Double
    Get
```

Property Value

Type: [Double](#)

The number of samples that have been acquired but not fetched for the record specified by record number to fetch.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionBinarySampleWidth Property

Gets or sets the bit width of the binary data in the acquired waveform, which is useful for determining which binary fetch method to use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long BinarySampleWidth { get; set; }
```

```
Public Property BinarySampleWidth As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value of this property is device-dependent. Valid values are 8, 16, and 32.

Remarks

To configure the device to store samples with a lower resolution than the native resolution, set this property to the desired binary width. This process can be useful for streaming at faster speeds at the cost of the resolution. The least significant bits are lost with this configuration.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferBandwidthMax Property

Gets or sets the maximum bandwidth that the device is allowed to consume.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DataTransferBandwidthMax { get; set; }
```

```
Public Property DataTransferBandwidthMax As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The maximum bandwidth on the PCIe bus that is used during data transfer.

Remarks

The NI device limits itself to transfer fewer bytes per second on the PCIe bus than the value you specify for this property.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferBlockSize Property

Gets or sets the maximum number of samples to transfer at one time from the device to host memory.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long DataTransferBlockSize { get; set; }
```

```
Public Property DataTransferBlockSize As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

Maximum number of samples to transfer at one time from the device to host memory.

Remarks

Increasing this number should result in better fetching performance because the driver does not need to restart the transfers as often. However, increasing this number may also increase the amount of page-locked memory required from the system.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDataTransferPacketSizePreferred Property

Gets or sets the preferred size of the data field in the PCI Express (PCIe) packet during data transfer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int DataTransferPacketSizePreferred { get; set; }
```

```
Public Property DataTransferPacketSizePreferred As Integer  
    Get  
    Set
```

Property Value

Type: [Int32](#)

This property must be a power of two (64, 128,... , 512).

Remarks

In general, the larger the packet size, the more efficiently the device uses the bus. However, some systems, because of their implementation, perform better with smaller packet sizes.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionDdcDataProcessingMode Property

Gets or sets the way in which data is processed by the digital downconverter (DDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public ScopeDdcDataProcessingMode DdcDataProcessingMode { get; set; }
```

```
Public Property DdcDataProcessingMode As ScopeDdcDataProcessingMode
    Get
    Set
```

Property Value

Type: [ScopeDdcDataProcessingMode](#)

Specifies the [ScopeDdcDataProcessingMode](#). The default value is [Complex](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with a device that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeDdcDataProcessingMode](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionEnabledChannels Property

Returns a comma-separated list of the channels enabled for the session in ascending order.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public string EnabledChannels { get; }
```

```
Public ReadOnly Property EnabledChannels As String
    Get
```

Property Value

Type: [String](#)

A comma-separated list of the channels enabled for the session in ascending order.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfMeasurementSamplesToFetch Property

Specifies the number of samples to fetch when performing a measurement. Use -1 to fetch the actual record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfMeasurementSamplesToFetch { get; set; }
```

```
Public Property NumberOfMeasurementSamplesToFetch As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is -1.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfPointsMin Property

Gets or sets the minimum number of points you require in the waveform record for each channel. NI-SCOPE uses the value you specify to configure the record length that the digitizer uses for waveform acquisition. [RecordLength](#) returns the actual record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfPointsMin { get; set; }
```

```
Public Property NumberOfPointsMin As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

Valid values range from 1 to the available onboard memory.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionNumberOfRecordsToFetch Property

Gets or sets the number of records to fetch.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfRecordsToFetch { get; set; }
```

```
Public Property NumberOfRecordsToFetch As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is -1.

Remarks

If you want to fetch all records from the record you specify in [RecordNumberToFetch](#) to the last record configured, use -1.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionOverflowErrorReportingMode Property

Gets or sets the reporting mode when the onboard signal processing block detects an overflow in any of its stages. Overflows lead to clipping of the waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeOverflowErrorReportingMode OverflowErrorReportingMode { get; set; }
```

```
Public Property OverflowErrorReportingMode As ScopeOverflowErrorReportingMode
    Get
    Set
```

Property Value

Type: [ScopeOverflowErrorReportingMode](#)

Specifies the [ScopeOverflowErrorReportingMode](#). The default value is [Warning](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI PXI-5142. If you open a session with a device that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionRecordLength Property

Gets the actual number of points the digitizer acquires for each channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordLength { get; }
```

```
Public ReadOnly Property RecordLength As Long  
    Get
```

Property Value

Type: [Int64](#)

The value is equal to or greater than the minimum number of points you specify with [NumberOfPointsMin](#).

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionRecordNumberToFetch Property

Gets or sets the record to fetch from a channel you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordNumberToFetch { get; set; }
```

```
Public Property RecordNumberToFetch As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 0. Valid values are greater than or equal to 0.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionResolution Property

Gets the bit width of valid data (as opposed to padding bits) in the acquired waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Resolution { get; }
```

```
Public ReadOnly Property Resolution As Long
    Get
```

Property Value

Type: [Int64](#)

The bit width of valid data in the acquired waveform.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleMode Property

Gets the [ScopeSampleMode](#) the digitizer is currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleMode SampleMode { get; }
```

```
Public ReadOnly Property SampleMode As ScopeSampleMode
    Get
```

Property Value

Type: [ScopeSampleMode](#)

Specifies the [ScopeSampleMode](#).

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleRate Property

Gets the effective sample rate using the current configuration in samples per second.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleRate { get; }
```

```
Public ReadOnly Property SampleRate As Double  
    Get
```

Property Value

Type: [Double](#)

The effective sample rate using the current configuration in samples per second.

Implements

[IIVI_ScopeAcquisitionSampleRate](#)

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionSampleRateMin Property

Gets or sets the sampling rate for the acquisition in samples per second.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleRateMin { get; set; }
```

```
Public Property SampleRateMin As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The combination of this property and [NumberOfPointsMin](#) must allow the digitizer to sample at a valid sampling rate and not require more memory than the onboard memory module allows.

Remarks

This property is the frequency at which digitized samples are stored, expressed in samples per second. This property is rounded up to the next valid sampling rate that the device supports. If [EnforceRealtime](#) is set to `True`, the sampling rate must be set lower than the [RealtimeSampleRateMax](#) of the digitizer, so data can be digitized at the requested rate. If [EnforceRealtime](#) is set to `False`, sampling rates can be set higher than the maximum real-time sampling rate of the digitizer. When this happens, NI-SCOPE enters random interleaved sampling (RIS) mode, where it acquires multiple waveforms at the maximum real-time sampling rate and reconstructs a periodic waveform.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionStartTime Property

Specifies the length of time, in seconds, from the trigger event to the first point in the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan StartTime { get; set; }
```

```
Public Property StartTime As PrecisionTimeSpan
    Get
    Set
```


Property Value

Type: PrecisionTimeSpan

Length of time from the trigger event to the first point in the waveform record in seconds.

Remarks

If the value is positive, the first point in the waveform record occurs after the trigger event. If the value is negative, the first point in the waveform record occurs before the trigger event.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionTimePerRecord Property

Gets or sets the length of time that corresponds to the record length.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimePerRecord { get; set; }
```

```
Public Property TimePerRecord As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: [PrecisionTimeSpan](#)

The length of time that corresponds to the record length.

See Also

Reference

[ScopeAcquisition Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionType Property

Specifies how the digitizer acquires data and fills the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public ScopeAcquisitionType Type { get; set; }
```

```
Public Property Type As ScopeAcquisitionType  
    Get  
    Set
```

Property Value

Type: [ScopeAcquisitionType](#)

Specifies the [ScopeAcquisitionType](#).

See Also

Reference

[ScopeAcquisition Class](#)





NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisition Methods

The ScopeAcquisition type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

ScopeAcquisition Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionStatus Enumeration

Specifies the status information about the acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeAcquisitionStatus
```

```
Public Enumeration ScopeAcquisitionStatus
```

Members

Member name	Value	Description
Complete	1	The acquisition is complete.
InProgress	0	The acquisition is in progress.
Unknown	-1	The acquisition status is unable to be retrieved.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAcquisitionType Enumeration

Specifies how the digitizer acquires data and fills the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeAcquisitionType
```

```
Public Enumeration ScopeAcquisitionType
```

Members

Member name	Value	Description
Normal	0	The normal resolution mode of the digitizer.
FlexibleResolution	1001	The flexible resolution mode of the digitizer. This value may not be supported on all devices.
Ddc	1002	The DDC mode of the NI 5620/5621 digitizer.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTrigger Class

Represents the ScopeAdvanceTrigger sub-objects that contain properties and methods related to the advance trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTrigger

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeAdvanceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeAdvanceTrigger
    Inherits ScopeSubObject
```






The ScopeAdvanceTrigger type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets the ScopeDigitalEdgeAdvanceTrigger sub-object that contains properties and methods that are related to digital edge advance triggers.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the advance trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction .
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTrigger Properties

The [ScopeAdvanceTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets the ScopeDigitalEdgeAdvanceTrigger sub-object that contains properties and methods that are related to digital edge advance triggers.

[Top](#)

See Also

Reference

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerDigitalEdge Property

Gets the [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeDigitalEdgeAdvanceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeAdvanceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeAdvanceTrigger](#)

The [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

See Also

Reference

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerDigitalEdge Property

Gets the [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeAdvanceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeAdvanceTrigger
    Get
```


Property Value

Type: [ScopeDigitalEdgeAdvanceTrigger](#)

The [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

See Also

Reference

[ScopeAdvanceTrigger Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTrigger Properties

The [ScopeAdvanceTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets the ScopeDigitalEdgeAdvanceTrigger sub-object that contains properties and methods that are related to digital edge advance triggers.

[Top](#)

See Also

Reference

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerDigitalEdge Property

Gets the [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeAdvanceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeAdvanceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeAdvanceTrigger](#)

The [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

See Also

Reference

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerDigitalEdge Property

Gets the [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeAdvanceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeAdvanceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeAdvanceTrigger](#)

The [ScopeDigitalEdgeAdvanceTrigger](#) sub-object that contains properties and methods that are related to digital edge advance triggers.

See Also

Reference

[ScopeAdvanceTrigger Class](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTrigger Methods

The [ScopeAdvanceTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the advance trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction .
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

[See Also](#)

[Reference](#)

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSendSoftwareEdgeTrigger Method

Sends the advance trigger to the digitizer. Call this method if [Source](#) has been configured to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

[See Also](#)

[Reference](#)

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[Other Resources](#)

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSendSoftwareEdgeTrigger Method

Sends the advance trigger to the digitizer. Call this method if [Source](#) has been configured to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Class

Represents the output terminal for [ScopeAdvanceTrigger](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerm

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


















C#VB






```
public sealed class ScopeAdvanceTriggerExportedOutputTerminal
```

```
Public NotInheritable Class ScopeAdvanceTriggerExportedOutputTerminal
```

The `ScopeAdvanceTriggerExportedOutputTerminal` type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.

 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)





Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeAdvanceTriggerExportedOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeAdvanceTriggerExportedOutputTerminal)	Determines whether the current instance of <code>ScopeAdvanceTriggerExportedOutputTerminal</code> and the specified <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeAdvanceTriggerExportedOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeAdvanceTriggerExportedOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

Name	Description
------	-------------

 <u>Equality</u>	Checks whether two <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object instances are equal.
 <u>(String to <code>ScopeAdvanceTriggerExportedOutputTerminal</code>)</u>	Converts a specified <code>String</code> to an equivalent <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object.
 <u>(<code>ScopeAdvanceTriggerExportedOutputTerminal</code> to String)</u>	Converts a <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object to an equivalent <code>String</code> .
 <u>Inequality</u>	Checks whether two <code>ScopeAdvanceTriggerExportedOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference























[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsAdvanceTriggerOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Properties

The [ScopeAdvanceTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)
 A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalClockC

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalNoSou

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi0 P

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi1 Pfi1

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi2

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPxiSta

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtSi0

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi1

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
ScopeAdvanceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi2

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtSi3

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal RtSi3 { get; }
```

```
Public Shared ReadOnly Property RtSi3 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi4

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi5

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi6

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi7

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSL_7".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)















[NationalInstruments.ModularInstruments.NIScope Namespace](#)









Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Properties

The [ScopeAdvanceTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.

 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0P

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
    ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```


Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
ScopeAdvanceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

A [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Methods

The [ScopeAdvanceTriggerExportedOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified object are equal.

		(Overrides ObjectEquals(Object) .)
⇒	Equals(ScopeAdvanceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified ScopeAdvanceTriggerExportedOutputTerminal object are equal.
⇒	FromString	Creates a ScopeAdvanceTriggerExportedOutputTerminal object from the specified String .
⇒	GetHashCode	Gets the hash code for the current instance of ScopeAdvanceTriggerExportedOutputTerminal . (Overrides ObjectGetHashCode .)
⇒	GetType	Gets the Type of the current instance. (Inherited from Object .)
⇒	ToString	Converts the current instance of ScopeAdvanceTriggerExportedOutputTerminal to String . (Overrides ObjectToString .)


[Top](#)[See Also](#)[Reference](#)[ScopeAdvanceTriggerExportedOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals

Overload List

Name	Description
⇒ Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)

 Equals(ScopeAdvanceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified ScopeAdvanceTriggerExportedOutputTerminal object are equal.
---	--

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified [ScopeAdvanceTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeAdvanceTriggerExportedOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeAdvanceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
 Specifies the [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals Method

Overload List

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeAdvanceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeAdvanceTriggerExportedOutputTerminal and the specified ScopeAdvanceTriggerExportedOutputTerminal object are equal.

[Top](#)[See Also](#)[Reference](#)[ScopeAdvanceTriggerExportedOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)


```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquals Method
(ScopeAdvanceTriggerExportedOutputTerminal)

Determines whether the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) and the specified [ScopeAdvanceTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeAdvanceTriggerExportedOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeAdvanceTriggerExportedOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
Specifies the [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalFromString Method

Creates a [ScopeAdvanceTriggerExportedOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerExportedOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeAdvanceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeAdvanceTriggerExportedOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

Returns an object of type [ScopeAdvanceTriggerExportedOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalGetHashCode Method

Gets the hash code for the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalToString Method

Converts the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Operators and Type Conversions

The [ScopeAdvanceTriggerExportedOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeAdvanceTriggerExportedOutputTerminal object instances are equal.

 (String to ScopeAdvanceTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeAdvanceTriggerExportedOutputTerminal object.
 (ScopeAdvanceTriggerExportedOutputTerminal to String)	Converts a ScopeAdvanceTriggerExportedOutputTerminal object to an equivalent String .
 Inequality	Checks whether two ScopeAdvanceTriggerExportedOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquality

Checks whether two [ScopeAdvanceTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeAdvanceTriggerExportedOutputTerminal source1,
    ScopeAdvanceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeAdvanceTriggerExportedOutputTerminal,
    source2 As ScopeAdvanceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
 Specifies the first [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
 Specifies the second [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalEquality Operator

Checks whether two [ScopeAdvanceTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeAdvanceTriggerExportedOutputTerminal source1,
    ScopeAdvanceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeAdvanceTriggerExportedOutputTerminal,
    source2 As ScopeAdvanceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
 Specifies the first [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
 Specifies the second [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also



Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Conversion Operators

Overload List

Name	Description
 (String to ScopeAdvanceTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeAdvanceTriggerExportedOutputTerminal object.
 (ScopeAdvanceTriggerExportedOutputTerminal to String)	Converts a ScopeAdvanceTriggerExportedOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Conversion (String to ScopeAdvanceTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeAdvanceTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static implicit operator ScopeAdvanceTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeAdvanceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeAdvanceTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

Returns the [ScopeAdvanceTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Conversion (String to ScopeAdvanceTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeAdvanceTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeAdvanceTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeAdvanceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeAdvanceTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

Returns the [ScopeAdvanceTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminal Conversion (ScopeAdvanceTriggerExportedOutputTerminal to String)

Converts a [ScopeAdvanceTriggerExportedOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeAdvanceTriggerExportedOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeAdvanceTriggerExportedOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
Specifies the [ScopeAdvanceTriggerExportedOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeAdvanceTriggerExportedOutputTerminal](#) object.

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

Overload

[NationalInstruments.ModularInstruments.NIScope](#) Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerExportedOutputTerminalInequality Operator

Checks whether two [ScopeAdvanceTriggerExportedOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeAdvanceTriggerExportedOutputTerminal source1,
    ScopeAdvanceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeAdvanceTriggerExportedOutputTerminal,
    source2 As ScopeAdvanceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
Specifies the first [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerExportedOutputTerminal](#)
Specifies the second [ScopeAdvanceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Class

Represents the source terminal for [ScopeDigitalEdgeAdvanceTrigger](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB














```
public sealed class ScopeAdvanceTriggerSource
```

```
Public NotInheritable Class ScopeAdvanceTriggerSource
```

The ScopeAdvanceTriggerSource type exposes the following members.







Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.

 Pfi0	Gets the source terminal for PFI 0.
 Pfi1	Gets the source terminal for PFI 1.
 Pfi2	Gets the source terminal for PFI 2.
 PxiStar	Gets the source terminal for PXI STAR.
 Rtsi0	Gets the source terminal for RTSI 0.
 Rtsi1	Gets the source terminal for RTSI 1.
 Rtsi2	Gets the source terminal for RTSI 2.
 Rtsi3	Gets the source terminal for RTSI 3.
 Rtsi4	Gets the source terminal for RTSI 4.
 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.





[Top](#)

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of <code>ScopeAdvanceTriggerSource</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeAdvanceTriggerSource)	Determines whether the current instance of <code>ScopeAdvanceTriggerSource</code> and the specified <code>ScopeAdvanceTriggerSource</code> object are equal.
	FromString	Creates a <code>ScopeAdvanceTriggerSource</code> object from the specified String .
	GetHashCode	Gets the hash code for the current instance of <code>ScopeAdvanceTriggerSource</code> . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of <code>ScopeAdvanceTriggerSource</code> to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether two <code>ScopeAdvanceTriggerSource</code> object instances are equal.
	(String to ScopeAdvanceTriggerSource)	Converts a specified <code>String</code> to an equivalent <code>ScopeAdvanceTriggerSource</code> object.
	(ScopeAdvanceTriggerSource to String)	Converts a <code>ScopeAdvanceTriggerSource</code> object to an equivalent <code>String</code> .
	Inequality	Checks whether two <code>ScopeAdvanceTriggerSource</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)















[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeAdvanceTriggerScopeDigitalEdgeAdvanceTriggerSource](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Properties

The [ScopeAdvanceTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.
	SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceSoftwareTriggerFunction

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As  
ScopeAdvanceTriggerSource  
Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)








Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Properties

The [ScopeAdvanceTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.

 Rtsi2	Gets the source terminal for RTSI 2.
 Rtsi3	Gets the source terminal for RTSI 3.
 Rtsi4	Gets the source terminal for RTSI 4.
 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeAdvanceTriggerSource  
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource Rtsi7 { get; }
```



```
Public Shared ReadOnly Property Rtsi7 As ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceSoftwareTriggerFunction Property

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As
    ScopeAdvanceTriggerSource
    Get
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

A [ScopeAdvanceTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Methods

The [ScopeAdvanceTriggerSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeAdvanceTriggerSource)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified ScopeAdvanceTriggerSource object are equal.
	FromString	Creates a ScopeAdvanceTriggerSource object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeAdvanceTriggerSource . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeAdvanceTriggerSource to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeAdvanceTriggerSource)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified ScopeAdvanceTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (ScopeAdvanceTriggerSource)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified [ScopeAdvanceTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeAdvanceTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeAdvanceTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the [ScopeAdvanceTriggerSource](#) object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeAdvanceTriggerSource)	Determines whether the current instance of ScopeAdvanceTriggerSource and the specified ScopeAdvanceTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquals Method (ScopeAdvanceTriggerSource)

Determines whether the current instance of [ScopeAdvanceTriggerSource](#) and the specified [ScopeAdvanceTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeAdvanceTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeAdvanceTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the [ScopeAdvanceTriggerSource](#) object to compare to the current instance of [ScopeAdvanceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceFromString Method

Creates a [ScopeAdvanceTriggerSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeAdvanceTriggerSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeAdvanceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeAdvanceTriggerSource](#) object returned represents.

Return Value

Type: [ScopeAdvanceTriggerSource](#)

Returns an object of type [ScopeAdvanceTriggerSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeAdvanceTriggerSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceToString Method

Converts the current instance of [ScopeAdvanceTriggerSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Operators and Type Conversions

The [ScopeAdvanceTriggerSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeAdvanceTriggerSource object instances are equal.
	(String to ScopeAdvanceTriggerSource)	Converts a specified String to an equivalent ScopeAdvanceTriggerSource object.
	(ScopeAdvanceTriggerSource to String)	Converts a ScopeAdvanceTriggerSource object to an equivalent String .
	Inequality	Checks whether two ScopeAdvanceTriggerSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquality Operator

Checks whether two [ScopeAdvanceTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeAdvanceTriggerSource source1,
    ScopeAdvanceTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeAdvanceTriggerSource,
    source2 As ScopeAdvanceTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the first [ScopeAdvanceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the second [ScopeAdvanceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceEquality Operator

Checks whether two [ScopeAdvanceTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeAdvanceTriggerSource source1,
    ScopeAdvanceTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeAdvanceTriggerSource,
    source2 As ScopeAdvanceTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the first [ScopeAdvanceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the second [ScopeAdvanceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeAdvanceTriggerSource)	Converts a specified String to an equivalent ScopeAdvanceTriggerSource object.
	(ScopeAdvanceTriggerSource to String)	Converts a ScopeAdvanceTriggerSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Conversion (String to ScopeAdvanceTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeAdvanceTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeAdvanceTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeAdvanceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeAdvanceTriggerSource](#) object.

Return Value

Type: [ScopeAdvanceTriggerSource](#)

Returns the [ScopeAdvanceTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Conversion (String to ScopeAdvanceTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeAdvanceTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeAdvanceTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeAdvanceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeAdvanceTriggerSource](#) object.

Return Value

Type: [ScopeAdvanceTriggerSource](#)

Returns the [ScopeAdvanceTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSource Conversion (ScopeAdvanceTriggerSource to String)

Converts a [ScopeAdvanceTriggerSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeAdvanceTriggerSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeAdvanceTriggerSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the [ScopeAdvanceTriggerSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeAdvanceTriggerSource](#) object.

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeAdvanceTriggerSourceInequality Operator

Checks whether two [ScopeAdvanceTriggerSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeAdvanceTriggerSource source1,  
    ScopeAdvanceTriggerSource source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeAdvanceTriggerSource,  
    source2 As ScopeAdvanceTriggerSource  
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the first [ScopeAdvanceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTriggerSource](#)
Specifies the second [ScopeAdvanceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeAdvanceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTrigger Class

Represents the `ScopeArmReferenceTrigger` sub-objects that contain properties and methods related to the arm reference trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTrigger](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax


C#VB

```
public sealed class ScopeArmReferenceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeArmReferenceTrigger
    Inherits ScopeSubObject
```






The `ScopeArmReferenceTrigger` type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets a ScopeDigitalEdgeArmReferenceTrigger sub-object that specifies properties and methods related to digital edge arm reference triggers.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the arm reference trigger to the digitizer. Call this method if you set Source to SoftwareTriggerFunction .
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

An analog edge trigger occurs when the trigger signal passes through the voltage threshold specified by [Level](#) and has the slope specified by [Slope](#).

For more information, refer to the Configure Trigger topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTrigger Properties

The [ScopeArmReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets a ScopeDigitalEdgeArmReferenceTrigger sub-object that specifies properties and methods related to digital edge arm reference triggers.

[Top](#)

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerDigitalEdge Property

Gets a [ScopeDigitalEdgeArmReferenceTrigger](#) sub-object that specifies properties and methods related to digital edge arm reference triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeArmReferenceTrigger DigitalEdge { get; }
```



```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeArmReferenceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeArmReferenceTrigger](#)

A sub-object of the type [ScopeDigitalEdgeArmReferenceTrigger](#).

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerDigitalEdge Property

Gets a [ScopeDigitalEdgeArmReferenceTrigger](#) sub-object that specifies properties and methods related to digital edge arm reference triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeArmReferenceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeArmReferenceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeArmReferenceTrigger](#)

A sub-object of the type [ScopeDigitalEdgeArmReferenceTrigger](#).

See Also

Reference

[ScopeArmReferenceTrigger Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTrigger Properties

The [ScopeArmReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets a ScopeDigitalEdgeArmReferenceTrigger sub-object that specifies properties and methods related to digital edge arm reference triggers.

[Top](#)

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerDigitalEdge Property

Gets a [ScopeDigitalEdgeArmReferenceTrigger](#) sub-object that specifies properties and methods related to digital edge arm reference triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeArmReferenceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeArmReferenceTrigger  
    Get
```

Property Value

Type: [ScopeDigitalEdgeArmReferenceTrigger](#)

A sub-object of the type [ScopeDigitalEdgeArmReferenceTrigger](#).

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerDigitalEdge Property

Gets a [ScopeDigitalEdgeArmReferenceTrigger](#) sub-object that specifies properties and methods related to digital edge arm reference triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeArmReferenceTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeArmReferenceTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeArmReferenceTrigger](#)

A sub-object of the type [ScopeDigitalEdgeArmReferenceTrigger](#).

See Also

Reference

[ScopeArmReferenceTrigger Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTrigger Methods

The [ScopeArmReferenceTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the arm reference trigger to the digitizer. Call this method if you set Source to SoftwareTriggerFunction .

 [ToString](#)

Returns a string that represents the current object.
(Inherited from [Object](#).)

[Top](#)

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSendSoftwareEdgeTrigge

Sends the arm reference trigger to the digitizer. Call this method if you set [Source](#) to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSendSoftwareEdgeTrigger Method

Sends the arm reference trigger to the digitizer. Call this method if you set [Source](#) to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Class

Represents the source terminal for [ScopeDigitalEdgeArmReferenceTrigger](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax















C#VB

```
public sealed class ScopeArmReferenceTriggerSource
```

```
Public NotInheritable Class ScopeArmReferenceTriggerSource
```

The ScopeArmReferenceTriggerSource type exposes the following members.







Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.
	SoftwareTriggerFunction	Gets the source terminal for the software trigger.

Top





Methods

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of <code>ScopeArmReferenceTriggerSource</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeArmReferenceTriggerSource)	Determines whether the current instance of <code>ScopeArmReferenceTriggerSource</code> and the specified <code>ScopeArmReferenceTriggerSource</code> object are equal.
 FromString	Creates a <code>ScopeArmReferenceTriggerSource</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeArmReferenceTriggerSource</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeArmReferenceTriggerSource</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether two <code>ScopeArmReferenceTriggerSource</code> object instances are equal.
	(String to ScopeArmReferenceTriggerSource)	Converts a specified <code>String</code> to an equivalent <code>ScopeArmReferenceTriggerSource</code> object.
	(ScopeArmReferenceTriggerSource to String)	Converts a <code>ScopeArmReferenceTriggerSource</code> object to an equivalent <code>String</code> .
	Inequality	Checks whether two <code>ScopeArmReferenceTriggerSource</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)











[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeArmReferenceTriggerScopeDigitalEdgeArmReferenceTriggerSource](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Properties

The [ScopeArmReferenceTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.

 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceSoftwareTriggerFu

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As  
ScopeArmReferenceTriggerSource  
Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)















[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Properties

The [ScopeArmReferenceTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.
	SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Immediate { get; }
```



```
Public Shared ReadOnly Property Immediate As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeArmReferenceTriggerSource
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeArmReferenceTriggerSource  
    Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceSoftwareTriggerFunction Property

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As  
ScopeArmReferenceTriggerSource  
Get
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

A [ScopeArmReferenceTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Methods

The [ScopeArmReferenceTriggerSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified object are equal. (Overrides Object.Equals(Object) .)
	Equals(ScopeArmReferenceTriggerSource)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified ScopeArmReferenceTriggerSource object are equal.

  FromString	Creates a ScopeArmReferenceTriggerSource object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeArmReferenceTriggerSource . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeArmReferenceTriggerSource to String . (Overrides ObjectToString .)

[Top](#)

See Also



Reference

[ScopeArmReferenceTriggerSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeArmReferenceTriggerSource)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified ScopeArmReferenceTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (ScopeArmReferenceTriggerSource)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified [ScopeArmReferenceTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeArmReferenceTriggerSource source
)
```

```
Public Function Equals (
    source As ScopeArmReferenceTriggerSource
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)

Specifies the [ScopeArmReferenceTriggerSource](#) object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)


[Equals Overload](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified object are equal.

	(Overrides ObjectEquals(Object).)
 Equals(ScopeArmReferenceTriggerSource)	Determines whether the current instance of ScopeArmReferenceTriggerSource and the specified ScopeArmReferenceTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquals Method (ScopeArmReferenceTriggerSource)

Determines whether the current instance of [ScopeArmReferenceTriggerSource](#) and the specified [ScopeArmReferenceTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeArmReferenceTriggerSource source
)
```

```
Public Function Equals (
    source As ScopeArmReferenceTriggerSource
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
 Specifies the [ScopeArmReferenceTriggerSource](#) object to compare to the current instance of [ScopeArmReferenceTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceFromString Method

Creates a [ScopeArmReferenceTriggerSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeArmReferenceTriggerSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeArmReferenceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeArmReferenceTriggerSource](#) object returned represents.

Return Value

Type: [ScopeArmReferenceTriggerSource](#)

Returns an object of type [ScopeArmReferenceTriggerSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeArmReferenceTriggerSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceToString Method

Converts the current instance of [ScopeArmReferenceTriggerSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Operators and Type Conversions

The [ScopeArmReferenceTriggerSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeArmReferenceTriggerSource object instances are equal.
	(String to ScopeArmReferenceTriggerSource)	Converts a specified String to an equivalent ScopeArmReferenceTriggerSource object.
	(ScopeArmReferenceTriggerSource to String)	Converts a ScopeArmReferenceTriggerSource object to an equivalent String .
	Inequality	Checks whether two ScopeArmReferenceTriggerSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquality Operator

Checks whether two [ScopeArmReferenceTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeArmReferenceTriggerSource source1,
    ScopeArmReferenceTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeArmReferenceTriggerSource,
    source2 As ScopeArmReferenceTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)

Specifies the first [ScopeArmReferenceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
 Specifies the second [ScopeArmReferenceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceEquality Operator

Checks whether two [ScopeArmReferenceTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static bool operator ==(
    ScopeArmReferenceTriggerSource source1,
    ScopeArmReferenceTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeArmReferenceTriggerSource,
    source2 As ScopeArmReferenceTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
Specifies the first [ScopeArmReferenceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
Specifies the second [ScopeArmReferenceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeArmReferenceTriggerSource)	Converts a specified String to an equivalent ScopeArmReferenceTriggerSource object.
	(ScopeArmReferenceTriggerSource to String)	Converts a ScopeArmReferenceTriggerSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Conversion (String to ScopeArmReferenceTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeArmReferenceTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeArmReferenceTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeArmReferenceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeArmReferenceTriggerSource](#) object.

Return Value

Type: [ScopeArmReferenceTriggerSource](#)

Returns the [ScopeArmReferenceTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Conversion (String to ScopeArmReferenceTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeArmReferenceTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static implicit operator ScopeArmReferenceTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeArmReferenceTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeArmReferenceTriggerSource](#) object.

Return Value

Type: [ScopeArmReferenceTriggerSource](#)

Returns the [ScopeArmReferenceTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSource Conversion (ScopeArmReferenceTriggerSource to String)

Converts a [ScopeArmReferenceTriggerSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeArmReferenceTriggerSource source  
)
```

```
Public Shared Widening Operator CType (
    source As ScopeArmReferenceTriggerSource
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
 Specifies the [ScopeArmReferenceTriggerSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeArmReferenceTriggerSource](#) object.

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArmReferenceTriggerSourceInequality Operator

Checks whether two [ScopeArmReferenceTriggerSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static bool operator !=(
    ScopeArmReferenceTriggerSource source1,
    ScopeArmReferenceTriggerSource source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeArmReferenceTriggerSource,
    source2 As ScopeArmReferenceTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
Specifies the first [ScopeArmReferenceTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTriggerSource](#)
Specifies the second [ScopeArmReferenceTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeArmReferenceTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeArrayMeasurementType Enumeration

Specifies various array measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeArrayMeasurementType
```

```
Public Enumeration ScopeArrayMeasurementType
```

Members

Member name	Value	Description
NoMeasurement	4000	No measurement is applied on the waveform.
LastAcquisitionHistogram	4001	The initialX value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The xIncrement returned is the bin size.
FftPhaseSpectrum	4002	The phase spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two. The phase spectrum is unwrapped.
FftAmplitudeSpectrumVoltsRms	4003	The amplitude spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two.
MultipleAcquisitionVoltageHistogram	4004	The initialX value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The xIncrement returned is the bin size.
MultipleAcquisitionTimeHistogram	4005	The initialX value returned is the bin value for the first bin, corresponding to the midpoint of the range of values in the first bin. The xIncrement returned is the bin size.
ArrayIntegral	4006	The waveform is integrated using Simpson's rule, fitting a parabola to every 3 points with the following equation:

		$y[2i+1] = (\text{waveform}[2i] + 4 * \text{waveform}[2i+1] + \text{waveform}[2i+2]) * dt / 3$
Derivative	4007	The differences in the waveform are computed using the formula: $y[i] = (\text{waveform}[i + 1] - \text{waveform}[i - 1]) / (2 * dt)$, where dt is the time between two points.
Inverse	4008	The inverse is computed using the following formula: $y[i] = 1.0 / \text{waveform}[i]$. If waveform[i] is zero, the inverse is zero.
HanningWindow	4009	A Hanning window is applied to the waveform using the following equation: $y[i] = 0.5 * \text{waveform}[i] * [1 - \cos(w)]$ where $w = (2)i/n$ and $n = \text{waveform size}$.
FlatTopWindow	4010	A Flat Top window is applied to the waveform using the following equation: $y[i] = \text{waveform}[i] * (0.2810639 - 0.5208972\cos(w) + 0.1980399\cos(2w))$ where $w = (2)i/n$ and n is the waveform size.
PolynomialInterpolation	4011	Polynomial interpolation allows oversampling or undersampling a waveform using any order polynomial set by PolynomialInterpolationOrder . For example, an order of 1 corresponds to linear interpolation. The new number of points is determined by the input size multiplied by InterpolationSamplingFactor . The sampling factor can be any number greater than 0.
MultiplyChannels	4012	Multiplies the waveforms from two channels, where one channel is specified by the parameter to FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo) , and the other channel is specified by OtherChannel . Any processing steps registered with AddWaveformProcessing(ScopeArrayMeasurementType) are completed for the "other channel" before this measurement is done. The two channels used in this measurement must be different.

AddChannels	4013	Adds the waveforms from two channels, where one channel is specified by the parameter to FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo) , and the other channel is specified by OtherChannel . Any processing steps registered with AddWaveformProcessing(ScopeArrayMeasurementType) are completed for the "other channel" before this measurement is done. The two channels used in this measurement must be different.
SubtractChannels	4014	Subtracts the waveforms from two channels, where one channel is specified by the parameter to FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo) , and the other channel is specified by OtherChannel . Any processing steps registered with AddWaveformProcessing(ScopeArrayMeasurementType) are completed for the "other channel" before this measurement is done. The two channels used in this measurement must be different.
DivideChannels	4015	Divides the waveforms from two channels, where one channel is specified by the parameter to FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo) and the other channel is specified by OtherChannel . Any processing steps registered with AddWaveformProcessing(ScopeArrayMeasurementType) are completed for the "other channel" before this measurement is done. The two channels used in this measurement must be different.
MultipleAcquisitionAverage	4016	The first time this measurement is called after it is cleared, an array the same size as the input is initialized to the input waveform, and the initialX and xIncrement values are set. Every subsequent call updates and returns the running average array without affecting the size of the array. The average array is cleared by calling ClearWaveformMeasurements with the measurement parameter set to MultipleAcquisitionAverage.

ButterworthFilter	4017	<p>The input waveform is filtered with a Butterworth IIR algorithm. The type of the filter is specified by <u>Type</u> as any of the following values: <u>LowPass</u>, <u>HighPass</u>, <u>BandPass</u>, or <u>BandStop</u>.</p> <p>The order of the filter is specified by <u>Order</u>. Also, a number of points equal to <u>TransientPercent</u> divided by 100 times the input size are eliminated from the beginning of the filtered waveform so the transient response of the filter does not affect further measurements.</p>
ChebyshevFilter	4018	<p>The input waveform is filtered with a Chebyshev IIR algorithm. The type of the filter is specified by <u>Type</u> as any of the following values: <u>LowPass</u>, <u>HighPass</u>, <u>BandPass</u>, or <u>BandStop</u>.</p> <p>The order of the filter is specified by <u>Order</u>. Also, a number of points equal to <u>TransientPercent</u> divided by 100 times the input size are eliminated from the beginning of the filtered waveform, so the transient response of the filter does not affect further measurements.</p>
FftAmplitudeSpectrumDB	4019	<p>The amplitude spectrum is calculated using a split-radix real FFT. If the number of acquired points is not a power of two, zeros are padded at the end of the waveform so it is the next higher power of two. The FFT is scaled to decibels:</p> $db[i] = 20.0 * \log_{10} (\text{Amplitude}[i] / \text{Peak Amplitude})$
HammingWindow	4020	<p>A Hamming window is applied to the waveform using the following equation:</p> $y[i] = x[i] [0.54 - 0.46\cos(w)]$ <p>where $w = (2 \pi) i/n$ and $n =$ the number of elements in x.</p>
WindowedFirFilter	4021	<p>FIR filters are well suited for applications that require phase information because the phase response is always linear. This prevents distortion of time-domain filtered data. The transient response of an FIR filter is a fixed number of points, filter taps - 1, which are eliminated in the filtered waveform returned.</p> <p>Windowed FIR filters use the cutoff frequency as part of the design process, but the cutoff is specified before the windowing operation. For Windowed FIR filters of</p>

		sufficient taps (more than 10 for a simple lowpass), the cutoff frequency has a magnitude near 0.5 (or 6 dB down). This varies slightly with the order and the window type.
BesselFilter	4022	The input waveform is filtered with a Bessel IIR algorithm. The type of the filter is specified by Type as any of the following values: LowPass , HighPass , BandPass , or BandStop . The order of the filter is specified by Order . Also, a number of points equal to TransientPercent divided by 100 times the input size are eliminated from the beginning of the filtered waveform so the transient response of the filter does not affect further measurements.
TriangleWindow	4023	A triangular window is applied to the waveform using the following equation: $y[i] = \text{waveform}[i] * (1 - w)$ where $w = (2i - n)/n$ and n is the number of elements in the waveform.
BlackmanWindow	4024	A Blackman window is applied to the waveform using the following equation: $y[i] = \text{waveform}[i] * (0.42659071 - 0.49656062\cos(w) + 0.07684867\cos(2w))$ where $w = (2i)/n$ and n is the number of elements in the waveform.
ArrayOffset	4025	The scalar value of ArrayOffset is added to every point in the waveform.
ArrayGain	4026	Every point in the waveform is multiplied by the scalar value ArrayGain .

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCableSenseMode Enumeration

Specifies the CableSense signal type

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeCableSenseMode
```

```
Public Enumeration ScopeCableSenseMode
```

Members

Member name	Value	Description
Disabled	0	The CableSense Signal is disabled
OnDemand	1	The CableSense signal with turn on/off when the CableSense Signal Start/Stop methods are invoked

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibration Class

Represents the ScopeCalibration sub-objects that contain properties and methods related to calibration.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeCalibration

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB

```
public sealed class ScopeCalibration : ScopeSubObject
```

```
Public NotInheritable Class ScopeCalibration
    Inherits ScopeSubObject
```




The ScopeCalibration type exposes the following members.


Properties

Name	Description
 External	Gets the ScopeExternalCalibration sub-object that provides properties and methods for external calibration.
 Self	Gets the ScopeSelfCalibration sub-object that provides properties and methods for self-calibration.
 Utility	Gets the ScopeCalibrationUtility sub-object that provides utility calibration methods.

Top

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance.

		(Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Calibration topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources




[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibration Properties

The [ScopeCalibration](#) type exposes the following members.

Properties

Name	Description
 External	Gets the ScopeExternalCalibration sub-object that provides properties and methods for external calibration.
 Self	Gets the ScopeSelfCalibration sub-object that provides properties and methods for self-calibration.
 Utility	Gets the ScopeCalibrationUtility sub-object that provides utility calibration methods.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationExternal Property

Gets the [ScopeExternalCalibration](#) sub-object that provides properties and methods for external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExternalCalibration External { get; }
```

```
Public ReadOnly Property External As ScopeExternalCalibration  
    Get
```

Property Value

Type: [ScopeExternalCalibration](#)

An object of type [ScopeExternalCalibration](#).

Remarks

External calibration is not supported for NI-SCOPE. This sub-object exposes only a sub-set of external calibration properties. For more information, refer to [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationExternal Property

Gets the [ScopeExternalCalibration](#) sub-object that provides properties and methods for external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExternalCalibration External { get; }
```

```
Public ReadOnly Property External As ScopeExternalCalibration  
    Get
```

Property Value

Type: [ScopeExternalCalibration](#)

An object of type [ScopeExternalCalibration](#).

Remarks

External calibration is not supported for NI-SCOPE. This sub-object exposes only a sub-set of external calibration properties. For more information, refer to [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationSelf Property

Gets the [ScopeSelfCalibration](#) sub-object that provides properties and methods for self-calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSelfCalibration Self { get; }
```

```
Public ReadOnly Property Self As ScopeSelfCalibration  
    Get
```

Property Value

Type: [ScopeSelfCalibration](#)

An object of type [ScopeSelfCalibration](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Property

Gets the [ScopeCalibrationUtility](#) sub-object that provides utility calibration methods.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCalibrationUtility Utility { get; }
```

```
Public ReadOnly Property Utility As ScopeCalibrationUtility  
    Get
```

Property Value

Type: [ScopeCalibrationUtility](#)

An object of type [ScopeCalibrationUtility](#).

See Also

Reference

[ScopeCalibration Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibration Properties

The [ScopeCalibration](#) type exposes the following members.

Properties

Name	Description
 External	Gets the ScopeExternalCalibration sub-object that provides properties and methods for external calibration.
 Self	Gets the ScopeSelfCalibration sub-object that provides properties and methods for self-calibration.
 Utility	Gets the ScopeCalibrationUtility sub-object that provides utility calibration methods.

[Top](#)

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationExternal Property

Gets the [ScopeExternalCalibration](#) sub-object that provides properties and methods for external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExternalCalibration External { get; }
```

```
Public ReadOnly Property External As ScopeExternalCalibration
    Get
```

Property Value

Type: [ScopeExternalCalibration](#)

An object of type [ScopeExternalCalibration](#).

Remarks

External calibration is not supported for NI-SCOPE. This sub-object exposes only a sub-set of external calibration properties. For more information, refer to [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationExternal Property

Gets the [ScopeExternalCalibration](#) sub-object that provides properties and methods for external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeExternalCalibration External { get; }
```

```
Public ReadOnly Property External As ScopeExternalCalibration  
    Get
```

Property Value

Type: [ScopeExternalCalibration](#)

An object of type [ScopeExternalCalibration](#).

Remarks

External calibration is not supported for NI-SCOPE. This sub-object exposes only a sub-set of external calibration properties. For more information, refer to [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationSelf Property

Gets the [ScopeSelfCalibration](#) sub-object that provides properties and methods for self-calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSelfCalibration Self { get; }
```

```
Public ReadOnly Property Self As ScopeSelfCalibration  
    Get
```

Property Value

Type: [ScopeSelfCalibration](#)

An object of type [ScopeSelfCalibration](#).

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Property

Gets the [ScopeCalibrationUtility](#) sub-object that provides utility calibration methods.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCalibrationUtility Utility { get; }
```

```
Public ReadOnly Property Utility As ScopeCalibrationUtility  
    Get
```

Property Value

Type: [ScopeCalibrationUtility](#)

An object of type [ScopeCalibrationUtility](#).

See Also

Reference

[ScopeCalibration Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibration Methods

The [ScopeCalibration](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Class

Represents the NI-SCOPE calibration utility.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeCalibrationUtility

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeCalibrationUtility : ScopeSubObject
```

```
Public NotInheritable Class ScopeCalibrationUtility
    Inherits ScopeSubObject
```





The ScopeCalibrationUtility type exposes the following members.

Properties

Name	Description
 DeviceTemperature	Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

Top

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Calibration topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Properties

The [ScopeCalibrationUtility](#) type exposes the following members.

Properties

Name	Description
 DeviceTemperature	Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

[Top](#)

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtilityDeviceTemperature Property

Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DeviceTemperature { get; }
```

```
Public ReadOnly Property DeviceTemperature As Double  
    Get
```

Property Value

Type: [Double](#)

The temperature of the instrument in degrees Celsius from the onboard sensor.

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtilityDeviceTemperature Property

Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DeviceTemperature { get; }
```

```
Public ReadOnly Property DeviceTemperature As Double  
    Get
```

Property Value

Type: [Double](#)

The temperature of the instrument in degrees Celsius from the onboard sensor.

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Properties

The [ScopeCalibrationUtility](#) type exposes the following members.

Properties

Name	Description
 DeviceTemperature	Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

[Top](#)

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtilityDeviceTemperature Property

Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DeviceTemperature { get; }
```

```
Public ReadOnly Property DeviceTemperature As Double
    Get
```

Property Value

Type: [Double](#)

The temperature of the instrument in degrees Celsius from the onboard sensor.

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtilityDeviceTemperature Property

Gets the temperature of the instrument in degrees Celsius from the onboard sensor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public double DeviceTemperature { get; }
```

```
Public ReadOnly Property DeviceTemperature As Double  
    Get
```

Property Value

Type: [Double](#)

The temperature of the instrument in degrees Celsius from the onboard sensor.

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCalibrationUtility Methods

The [ScopeCalibrationUtility](#) type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeCalibrationUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannel Class

Represents a channel or a set of channels and the properties that you can set on a channel.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeChannel

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax





C#VB


















```
public sealed class ScopeChannel : ScopeSubObject,
    IIVI_ScopeChannel, IIVI_RepeatedCapabilityIdentification
```

```
Public NotInheritable Class ScopeChannel
    Inherits ScopeSubObject
    Implements IIVI_ScopeChannel, IIVI_RepeatedCapabilityIdentification
```

The ScopeChannel type exposes the following members.










Properties

Name	Description
 BandpassFilterEnabled	Gets or sets whether the bandpass filter is enabled on the specified channel.
 CableSenseMode	Gets or sets the CableSense Signal Mode
 CableSenseVoltage	Gets the CableSense signal voltage amplitude for the given channel
 Coupling	Gets or sets how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

 DigitalGain	Gets or sets the gain on the specified channel in hardware before any onboard signal processing occurs.
 DigitalOffset	Gets or sets the offset on the specified channel in hardware before any onboard signal processing occurs.
 DitherEnabled	Gets or sets whether dither is enabled on the channel.
 Enabled	Gets or sets whether the digitizer acquires a waveform for the channel.
 EnableInterleavingOffsetCorrection	Gets or sets a value specifying whether to enable the interleaving offset correction.
 EnableTimeInterleavedSampling	Gets or sets the instrument to use multiple ADCs to sample the same waveform at a higher effective real-time sample rate.
 FlexFirAntiAliasFilterType	Gets or sets the type of filter needed to achieve desired filtering characteristics.
 HighPassFilterFrequency	Gets or sets the frequency for the highpass filter in Hz.
 InputFrequencyMax	Gets or sets the bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.
 InputImpedance	Gets or sets the input impedance, in ohms, for the channel.
 Measurement	Gets the ScopeChannelMeasurement sub-object that is related to configuring digitizers to obtain scalar and array measurements.
 Name	Gets the name of the channel.
 Offset	Gets or sets the location of the center of the range with respect to ground in volts.
 OnboardSignalProcessing	Gets the ScopeChannelOnboardSignalProcessing sub-object that is related to configuring the onboard signal processing (OSP) block of the high speed digitizer.
 ProbeAttenuation	Gets or sets the probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.
 Range	Gets or sets the absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.
 TerminalConfiguration	Gets or sets the terminal configuration for the channel.

[Top](#)

Methods

Name	Description
 Configure	Configures the most commonly-configured properties, such as the range, offset, coupling, probe attenuation, and the channel name of the digitizer vertical subsystem.
 ConfigureCharacteristics	Configures the properties that control the electrical characteristics of the channel.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetFrequencyResponse	Gets the frequency response of the digitizer for the current configurations of the channel attributes.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetNormalizationCoefficients	Returns coefficients that can be used to convert binary data to normalized and calibrated data.
 GetScalingCoefficients	Returns an array of ScopeCoefficientInfo containing gain and offset coefficients for a given channel.
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string representation of the object. (Overrides ObjectToString .)

[Top](#)

Remarks

For more information, refer to the Attributes section of the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources










[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)













Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannel Properties

The [ScopeChannel](#) type exposes the following members.

Properties

Name	Description
 BandpassFilterEnabled	Gets or sets whether the bandpass filter is enabled on the specified channel.
 CableSenseMode	Gets or sets the CableSense Signal Mode
 CableSenseVoltage	Gets the CableSense signal voltage amplitude for the given channel
 Coupling	Gets or sets how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.
 DigitalGain	Gets or sets the gain on the specified channel in hardware before any onboard signal processing occurs.
 DigitalOffset	Gets or sets the offset on the specified channel in hardware before any onboard signal processing occurs.
 DitherEnabled	Gets or sets whether dither is enabled on the channel.
 Enabled	Gets or sets whether the digitizer acquires a waveform for the channel.
 EnableInterleavingOffsetCorrection	Gets or sets a value specifying whether to enable the interleaving offset correction.

 EnableTimeInterleavedSampling	Gets or sets the instrument to use multiple ADCs to sample the same waveform at a higher effective real-time sample rate.
 FlexFirAntiAliasFilterType	Gets or sets the type of filter needed to achieve desired filtering characteristics.
 HighPassFilterFrequency	Gets or sets the frequency for the highpass filter in Hz.
 InputFrequencyMax	Gets or sets the bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.
 InputImpedance	Gets or sets the input impedance, in ohms, for the channel.
 Measurement	Gets the ScopeChannelMeasurement sub-object that is related to configuring digitizers to obtain scalar and array measurements.
 Name	Gets the name of the channel.
 Offset	Gets or sets the location of the center of the range with respect to ground in volts.
 OnboardSignalProcessing	Gets the ScopeChannelOnboardSignalProcessing sub-object that is related to configuring the onboard signal processing (OSP) block of the high speed digitizer.
 ProbeAttenuation	Gets or sets the probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.
 Range	Gets or sets the absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.
 TerminalConfiguration	Gets or sets the terminal configuration for the channel.

[Top](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelBandpassFilterEnabled Property

Gets or sets whether the bandpass filter is enabled on the specified channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool BandpassFilterEnabled { get; set; }
```

```
Public Property BandpassFilterEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelBandpassFilterEnabled Property

Gets or sets whether the bandpass filter is enabled on the specified channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool BandpassFilterEnabled { get; set; }
```

```
Public Property BandpassFilterEnabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCableSenseMode Property

Gets or sets the CableSense Signal Mode

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCableSenseMode CableSenseMode { get; set; }
```

```
Public Property CableSenseMode As ScopeCableSenseMode
    Get
    Set
```

Property Value

Type: [ScopeCableSenseMode](#)

Specifies the [ScopeCableSenseMode](#). The default value for this property is [Disabled](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCableSenseVoltage Property

Gets the CableSense signal voltage amplitude for the given channel

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CableSenseVoltage { get; set; }
```

```
Public Property CableSenseVoltage As Double
    Get
    Set
```

Property Value

Type: [Double](#)

A double that represents the CableSense signal voltage

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCoupling Property

Gets or sets how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeVerticalCoupling Coupling { get; set; }
```

```
Public Property Coupling As ScopeVerticalCoupling  
    Get  
    Set
```

Property Value

Type: [ScopeVerticalCoupling](#)

Specifies the [ScopeVerticalCoupling](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDigitalGain Property

Gets or sets the gain on the specified channel in hardware before any onboard signal processing occurs.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double DigitalGain { get; set; }
```

```
Public Property DigitalGain As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The gain can range from -1.5 to +1.5.

Remarks

If an overflow occurs, the data is clipped and the driver returns an error. To eliminate overflows, the digital gain should attenuate the ADC data, that is, ADC data should be set to a value less than 1.

The digital gain circuit can overflow if the following condition is not met:

$$-1 < \text{Unity Scaled User Data} \times \text{Pre-Filter Gain} < 1$$

You can also use Digital Gain to complete fine-gain adjustments on the ADC data to compensate for channel or system gain imperfections. These adjustments occur in hardware before the data is stored to memory. These adjustments save an additional processing step that normally occurs in software.

To change the default error reporting caused by OSP overflows, change the [OverflowErrorReportingMode](#) attribute.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[ScopeChannelDigitalOffset](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDigitalOffset Property

Gets or sets the offset on the specified channel in hardware before any onboard signal processing occurs.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public double DigitalOffset { get; set; }
```

```
Public Property DigitalOffset As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The valid values range from $-(\text{Vertical Range} \times 0.4)$ to $+(\text{Vertical Range} \times 0.4)$. The default value is 0.

Remarks

The digital offset circuit can overflow if the following condition is not met:

```
-(Vertical Range/2) <= Channel Data + Digital Offset <= +Vertical Range/2.
```

If an overflow occurs, the data is clipped and the driver returns an error. To change the default error reporting caused by OSP overflows, change the [OverflowErrorReportingMode](#) property.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI PXI-5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelDigitalGain](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDitherEnabled Property

Gets or sets whether dither is enabled on the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DitherEnabled { get; set; }
```

```
Public Property DitherEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

Using dither can improve the spectral performance of the instrument by reducing the effects of quantization. However, adding dither increases the power level to the ADC, so you may need to either decrease the signal level or increase the vertical range.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnabled Property

Gets or sets whether the digitizer acquires a waveform for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if the channel is enabled.

Implements

[IIVIChannelEnabled](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnableInterleavingOffsetCorrection Property

Gets or sets a value specifying whether to enable the interleaving offset correction.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool EnableInterleavingOffsetCorrection { get; set; }
```

```
Public Property EnableInterleavingOffsetCorrection As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, if interleaving offset correction is enabled; otherwise, . The default value is .

Exceptions

Exception	Condition
ObjectDisposedException	EnableInterleavingOffsetCorrection was called after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	On the instrument under use, EnableInterleavingOffsetCorrection is not supported.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Note

If disabled, warranted specifications are not guaranteed.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnableTimeInterleavedSampling Property

Gets or sets the instrument to use multiple ADCs to sample the same waveform at a higher effective real-time sample rate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool EnableTimeInterleavedSampling { get; set; }
```

```
Public Property EnableTimeInterleavedSampling As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is false.

Remarks

This property is not supported by all instruments. For more information, refer to the [Features Supported by Device](#) topic in the [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelFlexFirAntiAliasFilterType Property

Gets or sets the type of filter needed to achieve desired filtering characteristics.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFlexFirAntiAliasFilterType FlexFirAntiAliasFilterType { get; set; }
```

```
Public Property FlexFirAntiAliasFilterType As ScopeFlexFirAntiAliasFilterType  
    Get  
    Set
```

Property Value

Type: [ScopeFlexFirAntiAliasFilterType](#)

Specifies the [ScopeFlexFirAntiAliasFilterType](#). The default value for this property is [Tap48Standard](#).

Remarks

The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter. Use this property to select from several types of filters to achieve desired filtering characteristics. For most applications, the default value of this property is recommended. The other available filters are useful for optimizing settling time measurements of step responses.

Settling time values refer to the FIR filter only and do not take into account settling time caused by the analog front end. Refer to the NI PXI/PCI-5922 Specifications in the [NI High-Speed Digitizers Help](#) for combined digital and analog settling times.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelHighPassFilterFrequency Property

Gets or sets the frequency for the highpass filter in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighPassFilterFrequency { get; set; }
```

```
Public Property HighPassFilterFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The frequency for the highpass filter in Hz. The instrument uses one of the valid values listed in the remarks. If an invalid value is specified, no coercion occurs. The default value is 0.0.

Exceptions

Exception	Condition
ObjectDisposedException	HighPassFilterFrequency was called after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	On the instrument under use, HighPassFilterFrequency is not supported.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Valid values (PXIe-5164): 0.0, 90.0, 450.0

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelInputFrequencyMax Property

Gets or sets the bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double InputFrequencyMax { get; set; }
```

```
Public Property InputFrequencyMax As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value for this property is instrument-dependent.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelInputImpedance Property

Gets or sets the input impedance, in ohms, for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double InputImpedance { get; set; }
```

```
Public Property InputImpedance As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value for this property is instrument-dependent.

Implements

[IIVI_Scope_Channel_Input_Impedance](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Property

Gets the [ScopeChannelMeasurement](#) sub-object that is related to configuring digitizers to obtain scalar and array measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurement Measurement { get; }
```

```
Public ReadOnly Property Measurement As ScopeChannelMeasurement
    Get
```

Property Value

Type: [ScopeChannelMeasurement](#)

An object of type [ScopeChannelMeasurement](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelName Property

Gets the name of the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Name { get; }
```

```
Public ReadOnly Property Name As String
    Get
```

Property Value

Type: [String](#)

A string that represents the name of the channel.

Implements

[IIVIRepeatedCapabilityIdentificationName](#)

Exceptions

Exception	Condition
ObjectDisposedException	The object has been disposed.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOffset Property

Gets or sets the location of the center of the range with respect to ground in volts.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Offset { get; set; }
```

```
Public Property Offset As Double
    Get
    Set
```

Property Value

Type: [Double](#)

There is no default value for this property.

Implements

[IIVI_Scope_Channel_Offset](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Property

Gets the [ScopeChannelOnboardSignalProcessing](#) sub-object that is related to configuring the onboard signal processing (OSP) block of the high speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelOnboardSignalProcessing OnboardSignalProcessing { get; }
```

```
Public ReadOnly Property OnboardSignalProcessing As  
ScopeChannelOnboardSignalProcessing  
Get
```

Property Value

Type: [ScopeChannelOnboardSignalProcessing](#)

An object of type [ScopeChannelOnboardSignalProcessing](#).

Remarks

This property is valid only with high-speed digitizers, such as the NI PXI-5142, that support onboard signal processing (OSP).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelProbeAttenuation Property

Gets or sets the probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ProbeAttenuation { get; set; }
```

```
Public Property ProbeAttenuation As Double
    Get
    Set
```


Property Value

Type: [Double](#)

The probe attenuation for the input channel.

Implements

[IIVI_Scope_Channel_Probe_Attenuation](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelRange Property

Gets or sets the absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Range { get; set; }
```

```
Public Property Range As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The absolute value, in volts, of the input range for a channel.

Implements

[IIVI_Scope_Channel_Range](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelTerminalConfiguration Property

Gets or sets the terminal configuration for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelTerminalConfiguration TerminalConfiguration { get; set; }
```

```
Public Property TerminalConfiguration As ScopeChannelTerminalConfiguration  
    Get  
    Set
```

Property Value

Type: [ScopeChannelTerminalConfiguration](#)

Specifies the [ScopeChannelTerminalConfiguration](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources








[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannel Properties

The [ScopeChannel](#) type exposes the following members.

Properties

Name	Description
 BandpassFilterEnabled	Gets or sets whether the bandpass filter is enabled on the specified channel.
 CableSenseMode	Gets or sets the CableSense Signal Mode
 CableSenseVoltage	Gets the CableSense signal voltage amplitude for the given channel
 Coupling	Gets or sets how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.
 DigitalGain	Gets or sets the gain on the specified channel in hardware before any onboard signal processing occurs.
 DigitalOffset	Gets or sets the offset on the specified channel in hardware before any onboard signal processing occurs.
 DitherEnabled	Gets or sets whether dither is enabled on the channel.

 Enabled	Gets or sets whether the digitizer acquires a waveform for the channel.
 EnableInterleavingOffsetCorrection	Gets or sets a value specifying whether to enable the interleaving offset correction.
 EnableTimeInterleavedSampling	Gets or sets the instrument to use multiple ADCs to sample the same waveform at a higher effective real-time sample rate.
 FlexFirAntiAliasFilterType	Gets or sets the type of filter needed to achieve desired filtering characteristics.
 HighPassFilterFrequency	Gets or sets the frequency for the highpass filter in Hz.
 InputFrequencyMax	Gets or sets the bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.
 InputImpedance	Gets or sets the input impedance, in ohms, for the channel.
 Measurement	Gets the ScopeChannelMeasurement sub-object that is related to configuring digitizers to obtain scalar and array measurements.
 Name	Gets the name of the channel.
 Offset	Gets or sets the location of the center of the range with respect to ground in volts.
 OnboardSignalProcessing	Gets the ScopeChannelOnboardSignalProcessing sub-object that is related to configuring the onboard signal processing (OSP) block of the high speed digitizer.
 ProbeAttenuation	Gets or sets the probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.
 Range	Gets or sets the absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.
 TerminalConfiguration	Gets or sets the terminal configuration for the channel.

[Top](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelBandpassFilterEnabled Property

Gets or sets whether the bandpass filter is enabled on the specified channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool BandpassFilterEnabled { get; set; }
```

```
Public Property BandpassFilterEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelBandpassFilterEnabled Property

Gets or sets whether the bandpass filter is enabled on the specified channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool BandpassFilterEnabled { get; set; }
```

```
Public Property BandpassFilterEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCableSenseMode Property

Gets or sets the CableSense Signal Mode

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCableSenseMode CableSenseMode { get; set; }
```

```
Public Property CableSenseMode As ScopeCableSenseMode
    Get
    Set
```

Property Value

Type: [ScopeCableSenseMode](#)

Specifies the [ScopeCableSenseMode](#). The default value for this property is [Disabled](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCableSenseVoltage Property

Gets the CableSense signal voltage amplitude for the given channel

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CableSenseVoltage { get; set; }
```

```
Public Property CableSenseVoltage As Double
    Get
    Set
```

Property Value

Type: [Double](#)

A double that represents the CableSense signal voltage

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCoupling Property

Gets or sets how the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeVerticalCoupling Coupling { get; set; }
```

```
Public Property Coupling As ScopeVerticalCoupling  
    Get  
    Set
```

Property Value

Type: [ScopeVerticalCoupling](#)

Specifies the [ScopeVerticalCoupling](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDigitalGain Property

Gets or sets the gain on the specified channel in hardware before any onboard signal processing occurs.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public double DigitalGain { get; set; }
```

```
Public Property DigitalGain As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The gain can range from -1.5 to +1.5.

Remarks

If an overflow occurs, the data is clipped and the driver returns an error. To eliminate overflows, the digital gain should attenuate the ADC data, that is, ADC data should be set to a value less than 1.

The digital gain circuit can overflow if the following condition is not met:

```
-1 < Unity Scaled User Data × Pre-Filter Gain < 1
```

You can also use Digital Gain to complete fine-gain adjustments on the ADC data to compensate for channel or system gain imperfections. These adjustments occur in hardware before the data is stored to memory. These adjustments save an additional processing step that normally occurs in software.

To change the default error reporting caused by OSP overflows, change the [OverflowErrorReportingMode](#) attribute.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[ScopeChannelDigitalOffset](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDigitalOffset Property

Gets or sets the offset on the specified channel in hardware before any onboard signal processing occurs.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public double DigitalOffset { get; set; }
```

```
Public Property DigitalOffset As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The valid values range from $-(\text{Vertical Range} \times 0.4)$ to $+(\text{Vertical Range} \times 0.4)$. The default value is 0.

Remarks

The digital offset circuit can overflow if the following condition is not met:

```
-(Vertical Range/2) <= Channel Data + Digital Offset <= +Vertical Range/2.
```

If an overflow occurs, the data is clipped and the driver returns an error. To change the default error reporting caused by OSP overflows, change the [OverflowErrorReportingMode](#) property.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI PXI-5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

[ScopeChannelDigitalGain](#)

[ScopeChannelOnboardSignalProcessing](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDitherEnabled Property

Gets or sets whether dither is enabled on the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DitherEnabled { get; set; }
```

```
Public Property DitherEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

Using dither can improve the spectral performance of the instrument by reducing the effects of quantization. However, adding dither increases the power level to the ADC, so you may need to either decrease the signal level or increase the vertical range.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnabled Property

Gets or sets whether the digitizer acquires a waveform for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if the channel is enabled.

Implements

[IIVIChannelEnabled](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnableInterleavingOffsetCorrection Property

Gets or sets a value specifying whether to enable the interleaving offset correction.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool EnableInterleavingOffsetCorrection { get; set; }
```

```
Public Property EnableInterleavingOffsetCorrection As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, if interleaving offset correction is enabled; otherwise, . The default value is .

Exceptions

Exception	Condition
ObjectDisposedException	EnableInterleavingOffsetCorrection was called after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	On the instrument under use, EnableInterleavingOffsetCorrection is not supported.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Note

If disabled, warranted specifications are not guaranteed.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEnableTimeInterleavedSampling Property

Gets or sets the instrument to use multiple ADCs to sample the same waveform at a higher effective real-time sample rate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool EnableTimeInterleavedSampling { get; set; }
```

```
Public Property EnableTimeInterleavedSampling As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is false.

Remarks

This property is not supported by all instruments. For more information, refer to the Features Supported by Device topic in the [NI High-Speed Digitizers Help](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelFlexFirAntiAliasFilterType Property

Gets or sets the type of filter needed to achieve desired filtering characteristics.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFlexFirAntiAliasFilterType FlexFirAntiAliasFilterType { get; set; }
```

```
Public Property FlexFirAntiAliasFilterType As ScopeFlexFirAntiAliasFilterType  
    Get  
    Set
```

Property Value

Type: [ScopeFlexFirAntiAliasFilterType](#)

Specifies the [ScopeFlexFirAntiAliasFilterType](#). The default value for this property is [Tap48Standard](#).

Remarks

The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter. Use this property to select from several types of filters to achieve desired filtering characteristics. For most applications, the default value of this property is recommended. The other available filters are useful for optimizing settling time measurements of step responses.

Settling time values refer to the FIR filter only and do not take into account settling time caused by the analog front end. Refer to the NI PXI/PCI-5922 Specifications in the [NI High-Speed Digitizers Help](#) for combined digital and analog settling times.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelHighPassFilterFrequency Property

Gets or sets the frequency for the highpass filter in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighPassFilterFrequency { get; set; }
```

```
Public Property HighPassFilterFrequency As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The frequency for the highpass filter in Hz. The instrument uses one of the valid values listed in the remarks. If an invalid value is specified, no coercion occurs. The default value is 0.0.

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	HighPassFilterFrequency was called after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	On the instrument under use, HighPassFilterFrequency is not supported.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Valid values (PXIe-5164): 0.0, 90.0, 450.0

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelInputFrequencyMax Property

Gets or sets the bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double InputFrequencyMax { get; set; }
```

```
Public Property InputFrequencyMax As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value for this property is instrument-dependent.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelInputImpedance Property

Gets or sets the input impedance, in ohms, for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public double InputImpedance { get; set; }
```

```
Public Property InputImpedance As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value for this property is instrument-dependent.

Implements

[IIVI_Scope_Channel_Input_Impedance](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Property

Gets the [ScopeChannelMeasurement](#) sub-object that is related to configuring digitizers to obtain scalar and array measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurement Measurement { get; }
```

```
Public ReadOnly Property Measurement As ScopeChannelMeasurement  
    Get
```

Property Value

Type: [ScopeChannelMeasurement](#)

An object of type [ScopeChannelMeasurement](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelName Property

Gets the name of the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string Name { get; }
```

```
Public ReadOnly Property Name As String
    Get
```

Property Value

Type: [String](#)

A string that represents the name of the channel.

Implements

[IIVIRepeatedCapabilityIdentificationName](#)

Exceptions

Exception	Condition
ObjectDisposedException	The object has been disposed.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOffset Property

Gets or sets the location of the center of the range with respect to ground in volts.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public double Offset { get; set; }
```

```
Public Property Offset As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

There is no default value for this property.

Implements

[IIVI_ScopeChannelOffset](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Property

Gets the [ScopeChannelOnboardSignalProcessing](#) sub-object that is related to configuring the onboard signal processing (OSP) block of the high speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelOnboardSignalProcessing OnboardSignalProcessing { get; }
```

```
Public ReadOnly Property OnboardSignalProcessing As  
ScopeChannelOnboardSignalProcessing  
Get
```

Property Value

Type: [ScopeChannelOnboardSignalProcessing](#)

An object of type [ScopeChannelOnboardSignalProcessing](#).

Remarks

This property is valid only with high-speed digitizers, such as the NI PXI-5142, that support onboard signal processing (OSP).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelProbeAttenuation Property

Gets or sets the probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ProbeAttenuation { get; set; }
```

```
Public Property ProbeAttenuation As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The probe attenuation for the input channel.

Implements

[IIVI_ScopeChannelProbeAttenuation](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelRange Property

Gets or sets the absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Range { get; set; }
```

```
Public Property Range As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The absolute value, in volts, of the input range for a channel.

Implements

[IIVI_ScopeChannelRange](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelTerminalConfiguration Property

Gets or sets the terminal configuration for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelTerminalConfiguration TerminalConfiguration { get; set; }
```

```
Public Property TerminalConfiguration As ScopeChannelTerminalConfiguration  
    Get  
    Set
```

Property Value

Type: [ScopeChannelTerminalConfiguration](#)
Specifies the [ScopeChannelTerminalConfiguration](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources










[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannel Methods

The [ScopeChannel](#) type exposes the following members.

Methods

Name	Description
 Configure	Configures the most commonly-configured properties, such as the range, offset, coupling, probe attenuation, and the channel name of the digitizer vertical subsystem.
 ConfigureCharacteristics	Configures the properties that control the electrical characteristics of the channel.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetFrequencyResponse	Gets the frequency response of the digitizer for the current configurations of the channel attributes.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetNormalizationCoefficients	Returns coefficients that can be used to convert binary data to normalized and calibrated data.
 GetScalingCoefficients	Returns an array of ScopeCoefficientInfo containing gain and offset coefficients for a given channel.
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string representation of the object. (Overrides ObjectToString .)

[Top](#)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelConfigure Method

Configures the most commonly-configured properties, such as the range, offset, coupling, probe attenuation, and the channel name of the digitizer vertical subsystem.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    double verticalRange,  
    double verticalOffset,  
    ScopeVerticalCoupling verticalCoupling,  
    double probeAttenuation,  
    bool enabled  
)
```

```
Public Sub Configure (  
    verticalRange As Double,  
    verticalOffset As Double,  
    verticalCoupling As ScopeVerticalCoupling,  
    probeAttenuation As Double,  
    enabled As Boolean  
)
```

Parameters

verticalRange

Type: [SystemDouble](#)

The absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.

verticalOffset

Type: [SystemDouble](#)

The location of the center of the range, in volts, with respect to ground.

verticalCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeVerticalCoupling](#)

The way the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

probeAttenuation

Type: [SystemDouble](#)

The probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.

enabled

Type: [SystemBoolean](#)

A value indicating whether the digitizer acquires a waveform for the channel.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelConfigure Method

Configures the most commonly-configured properties, such as the range, offset, coupling, probe attenuation, and the channel name of the digitizer vertical subsystem.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    double verticalRange,
    double verticalOffset,
    ScopeVerticalCoupling verticalCoupling,
    double probeAttenuation,
```

```

        bool enabled
    )

```

```

Public Sub Configure (
    verticalRange As Double,
    verticalOffset As Double,
    verticalCoupling As ScopeVerticalCoupling,
    probeAttenuation As Double,
    enabled As Boolean
)

```

Parameters

verticalRange

Type: [SystemDouble](#)

The absolute value, in volts, of the input range for a channel. For example, to acquire a sine wave that spans between -5 and +5 V, set this property to 10.0 V.

verticalOffset

Type: [SystemDouble](#)

The location of the center of the range, in volts, with respect to ground.

verticalCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeVerticalCoupling](#)

The way the digitizer couples the input signal for the channel. When input coupling changes, the input stage takes a finite amount of time to settle.

probeAttenuation

Type: [SystemDouble](#)

The probe attenuation for the input channel. For example, for a 10:1 probe, set this property to 10.0.

enabled

Type: [SystemBoolean](#)

A value indicating whether the digitizer acquires a waveform for the channel.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelConfigureCharacteristics Method

Configures the properties that control the electrical characteristics of the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ConfigureCharacteristics(  
    double inputImpedance,  
    double inputFrequencyMax  
)
```

```
Public Sub ConfigureCharacteristics (  
    inputImpedance As Double,  
    inputFrequencyMax As Double  
)
```

Parameters

inputImpedance

Type: [SystemDouble](#)

Input impedance, in ohms, for the channel.

inputFrequencyMax

Type: [SystemDouble](#)

Bandwidth, in hertz, of the channel. Express this value as the frequency at which the input circuitry attenuates the input signal by 3 dB.

Implements

IIviScopeChannelConfigureCharacteristics(Double, Double)

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelGetFrequencyResponse Method

Gets the frequency response of the digitizer for the current configurations of the channel attributes.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void GetFrequencyResponse(  
    out double[] frequencies,  
    out double[] amplitudes,  
    out double[] phases  
)
```

```
Public Sub GetFrequencyResponse (  
    <OutAttribute> ByRef frequencies As Double(),  
    <OutAttribute> ByRef amplitudes As Double(),  
    <OutAttribute> ByRef phases As Double()  
)
```

Parameters

frequencies

Type: [SystemDouble](#)

The frequencies that correspond with the amplitude and phase response of the instrument.

amplitudes

Type: [SystemDouble](#)

The magnitudes that correspond with the magnitude response of the instrument.

phases

Type: [SystemDouble](#)

The phases that correspond with the phase response of the instrument.

Remarks

You can use this method only with high-speed digitizers that support onboard signal processing (OSP).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelGetNormalizationCoefficients Method

Returns coefficients that can be used to convert binary data to normalized and calibrated data.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCoefficientInfo[] GetNormalizationCoefficients()
```

```
Public Function GetNormalizationCoefficients As ScopeCoefficientInfo()
```

Return Value

Type: [ScopeCoefficientInfo](#)

The number of coefficient sets for this channel.

Remarks

Refer to the [Scaling and Normalization of Binary Data](#) for more information about how to use this method.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelGetScalingCoefficients Method

Returns an array of [ScopeCoefficientInfo](#) containing gain and offset coefficients for a given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeCoefficientInfo[] GetScalingCoefficients()
```

```
Public Function GetScalingCoefficients As ScopeCoefficientInfo()
```

Return Value

Type: [ScopeCoefficientInfo](#)

The number of coefficient sets for this channel.

Remarks

Refer to the [Scaling and Normalization of Binary Data](#) for more information about how to use this method.

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelToString Method

Returns a string representation of the object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

A string representation of the object.

Remarks

Overrides [ToString](#).

See Also

Reference

[ScopeChannel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollection Class

Represents a strongly-typed collection of ScopeChannelCollection objects.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeChannelCollection

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public sealed class ScopeChannelCollection : ScopeSubObject,
    IIVI_ScopeChannelCollection,
    IIVI_RepeatedCapabilityCollection<IIVI_ScopeChannel>,
    IEnumerable<IIVI_ScopeChannel>,
    IEnumerable, IIVI_RepeatedCapabilityCollection<ScopeChannel>,
    IEnumerable<ScopeChannel>
```

```
Public NotInheritable Class ScopeChannelCollection
    Inherits ScopeSubObject
    Implements IIVI_ScopeChannelCollection,
    IIVI_RepeatedCapabilityCollection(Of IIVI_ScopeChannel),
    IEnumerable(Of IIVI_ScopeChannel), IEnumerable,
    IIVI_RepeatedCapabilityCollection(Of ScopeChannel),
    IEnumerable(Of ScopeChannel)
```






The `ScopeChannelCollection` type exposes the following members.

Properties

Name	Description
 Count	Gets the number of cached ScopeChannel objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").
 Item	Gets a ScopeChannel object from the collection with the specified name. In Visual C#, this property is the indexer. <ul style="list-style-type: none"> ▪ If the channel name is a string specifying a list or range of channels (e.g., "0,1,2" or "0-6"), a new ScopeChannel is created and returned. The ScopeChannel created is not added to the collection. ▪ If the channel name corresponds to only one channel (e.g., "0" or "1") and if the channel has previously been accessed and cached in the collection, the corresponding ScopeChannel is returned; otherwise if the specified channel name exists on the instrument, a new ScopeChannel object is created, added to the collection, and returned.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the collection of single-channel ScopeChannel instances.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Attributes section in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollection Properties

The [ScopeChannelCollection](#) type exposes the following members.

Properties

Name	Description
 Count	Gets the number of cached ScopeChannel objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").
 Item	<p>Gets a ScopeChannel object from the collection with the specified name. In Visual C#, this property is the indexer.</p> <ul style="list-style-type: none"> ▪ If the channel name is a string specifying a list or range of channels (e.g., "0,1,2" or "0-6"), a new ScopeChannel is created and returned. The ScopeChannel created is not added to the collection. ▪ If the channel name corresponds to only one channel (e.g., "0" or "1") and if the channel has previously been accessed and cached in the collection, the corresponding ScopeChannel is returned; otherwise if the specified channel name exists on the instrument, a new ScopeChannel object is created, added to the collection, and returned.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionCount Property

Gets the number of cached [ScopeChannel](#) objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Count { get; }
```

```
Public ReadOnly Property Count As Integer  
    Get
```

Property Value

Type: [Int32](#)

The number of cached [ScopeChannel](#) objects contained in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Implements

[IIviRepeatedCapabilityCollectionTCount](#)

[IIviRepeatedCapabilityCollectionTCount](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionCount Property

Gets the number of cached [ScopeChannel](#) objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Count { get; }
```

```
Public ReadOnly Property Count As Integer  
    Get
```

Property Value

Type: [Int32](#)

The number of cached [ScopeChannel](#) objects contained in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Implements

[IIviRepeatedCapabilityCollectionTCount](#)

[IIviRepeatedCapabilityCollectionTCount](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionItem Property

Gets a [ScopeChannel](#) object from the collection with the specified name. In Visual C#, this property is the indexer.

- If the channel name is a string specifying a list or range of channels (e.g., "0,1,2" or "0-6"), a new [ScopeChannel](#) is created and returned. The [ScopeChannel](#) created is not added to the collection.
- If the channel name corresponds to only one channel (e.g., "0" or "1") and if the channel has previously been accessed and cached in the collection, the corresponding [ScopeChannel](#) is returned; otherwise if the specified channel name exists on the instrument, a new [ScopeChannel](#) object is created, added to the collection, and returned.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannel this[
    string name
] { get; }
```

```
Public ReadOnly Default Property Item (
    name As String
```

```
) As ScopeChannel
    Get
```

Parameters

name

Type: [SystemString](#)

A string specifying the name of a single channel (e.g., "0"), a comma-separated list of channels (e.g., "0,1,2"), or range of channels (e.g., "0-2").

Property Value

Type: [ScopeChannel](#)

The [ScopeChannel](#) corresponding to the name of the channel.

Implements

[IIVIRepeatedCapabilityCollectionTItemString](#)

See Also

Reference

[ScopeChannelCollection Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollection Properties

The [ScopeChannelCollection](#) type exposes the following members.

Properties

Name	Description
 Count	Gets the number of cached ScopeChannel objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").
 Item	Gets a ScopeChannel object from the collection with the specified name. In Visual C#, this property is the indexer.

- If the channel name is a string specifying a list or range of channels (e.g., "0,1,2" or "0-6"), a new [ScopeChannel](#) is created and returned. The [ScopeChannel](#) created is not added to the collection.
- If the channel name corresponds to only one channel (e.g., "0" or "1") and if the channel has previously been accessed and cached in the collection, the corresponding [ScopeChannel](#) is returned; otherwise if the specified channel name exists on the instrument, a new [ScopeChannel](#) object is created, added to the collection, and returned.

[Top](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionCount Property

Gets the number of cached [ScopeChannel](#) objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Count { get; }
```

```
Public ReadOnly Property Count As Integer
    Get
```

Property Value

Type: [Int32](#)

The number of cached [ScopeChannel](#) objects contained in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Implements

[IIVIRepeatedCapabilityCollectionTCount](#)

[IIVIRepeatedCapabilityCollectionTCount](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionCount Property

Gets the number of cached [ScopeChannel](#) objects in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Count { get; }
```

```
Public ReadOnly Property Count As Integer  
    Get
```

Property Value

Type: [Int32](#)

The number of cached [ScopeChannel](#) objects contained in the collection, which is equal to the number of channels configured using a single-channel string (e.g., "0" or "1").

Implements

[IIVIRepeatedCapabilityCollectionTCCount](#)

[IIVIRepeatedCapabilityCollectionTCCount](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionItem Property

Gets a [ScopeChannel](#) object from the collection with the specified name. In Visual C#, this property is the indexer.

- If the channel name is a string specifying a list or range of channels (e.g., "0,1,2" or "0-6"), a new [ScopeChannel](#) is created and returned. The [ScopeChannel](#) created is not added to the collection.
- If the channel name corresponds to only one channel (e.g., "0" or "1") and if the channel has previously been accessed and cached in the collection, the corresponding [ScopeChannel](#) is returned; otherwise if the specified channel name exists on the instrument, a new [ScopeChannel](#) object is created, added to the collection, and returned.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in

[NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public ScopeChannel this[
    string name
] { get; }
```

```
Public ReadOnly Default Property Item (
    name As String
) As ScopeChannel
    Get
```

Parameters

name

Type: [SystemString](#)

A string specifying the name of a single channel (e.g., "0"), a comma-separated list of channels (e.g., "0,1,2"), or range of channels (e.g., "0-2").

Property Value

Type: [ScopeChannel](#)

The [ScopeChannel](#) corresponding to the name of the channel.

Implements

[IIVIRepeatedCapabilityCollectionTItemString](#)

See Also

Reference

[ScopeChannelCollection Class](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollection Methods

The [ScopeChannelCollection](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetEnumerator	Returns an enumerator that iterates through the collection of single-channel ScopeChannel instances.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionGetEnumerator Method

Returns an enumerator that iterates through the collection of single-channel [ScopeChannel](#) instances.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IEnumerable<ScopeChannel> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerable(Of ScopeChannel)
```

Return Value

Type: [IEnumerableScopeChannel](#)

An enumerator that iterates through the collection of single-channel [ScopeChannel](#) instances.

Implements

[IEnumerableTGetEnumerator](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelCollectionGetEnumerator Method

Returns an enumerator that iterates through the collection of single-channel [ScopeChannel](#) instances.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IEnumerable<ScopeChannel> GetEnumerator()
```



```
Public Function GetEnumerator As IEnumerable(Of ScopeChannel)
```

Return Value

Type: [IEnumerableScopeChannel](#)

An enumerator that iterates through the collection of single-channel [ScopeChannel](#) instances.

Implements

[IEnumerableTGetEnumerator](#)

See Also

Reference

[ScopeChannelCollection Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdc Class

Defines the properties related to the digital down converter (DDC) block in a high speed digitizer.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax







C#VB

```
public sealed class ScopeChannelDdc : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelDdc
    Inherits ScopeSubObject
```





The ScopeChannelDdc type exposes the following members.

Properties

Name	Description
 CenterFrequency	Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.
 Enabled	Gets or sets whether the digital down converter (DDC) block of the digitizer is enabled.
 FrequencyTranslationEnabled	Gets or sets whether the frequency translation of the data around the selected center frequency down to baseband is enabled.
 FrequencyTranslationPhaseI	Gets or sets the I oscillator phase in degrees at the first point acquired.
 FrequencyTranslationPhaseQ	Gets or sets the Q oscillator phase in degrees at the first point acquired.
 QSource	Gets or sets the channel that is the input to the Q data stream of the digital downconverter (DDC).

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

You can only query or set the properties of this class when the scope session is associated with an instrument that supports onboard signal processing (OSP). You can use the DDC block of the NI 5142 for the following applications:

- **Alias-Protected Decimation:** For alias-protected decimation applications, complete the following steps:
 1. Set `DDC Enabled` to `.`
 2. Set `DdcDataProcessingMode` to `Real`.
 3. Set `SampleRateMin` to the desired decimated sample rate.
 4. Set `FrequencyTranslationEnabled` to `.`

- **AM Demodulation:** For AM demodulation applications, complete the following steps:
 1. Set `DDC Enabled` to `.`
 2. Set `DdcDataProcessingMode` to `Real`.
 3. Set `SampleRateMin` to the desired baseband sample rate.
 4. Set `FrequencyTranslationEnabled` to `.`
 5. Set `CenterFrequency` to the carrier frequency of the AM signal you want to downconvert.

- **Quadrature Downconversion:** For quadrature downconversion applications, complete the following steps:
 1. Set `DDC Enabled` to `.`
 2. Set `DdcDataProcessingMode` to `Complex`.
 3. Set `SampleRateMin` to the desired IQ data rate.
 4. Set `FrequencyTranslationEnabled` to `.`
 5. Set `CenterFrequency` to the desired center frequency of the IF band you want to downconvert.

- Baseband Decimation: For baseband decimation applications, complete the following steps:
 1. Set DDC [Enabled](#) to .
 2. Set [DdcDataProcessingMode](#) to [Complex](#).
 3. Set [SampleRateMin](#) to the desired baseband sample rate.
 4. Set [FrequencyTranslationEnabled](#) to .
 5. Set [QSource](#) to channel "1."

To use the NI 5142 in normal digitizer mode, set [Enabled](#) to .

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)


[NationalInstruments.ModularInstruments.NIScopeScopeChannelOnboardSignalProcessing](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdc Properties

The [ScopeChannelDdc](#) type exposes the following members.

Properties

Name	Description
 CenterFrequency	Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.

 Enabled	Gets or sets whether the digital down converter (DDC) block of the digitizer is enabled.
 FrequencyTranslationEnabled	Gets or sets whether the frequency translation of the data around the selected center frequency down to baseband is enabled.
 FrequencyTranslationPhaseI	Gets or sets the I oscillator phase in degrees at the first point acquired.
 FrequencyTranslationPhaseQ	Gets or sets the Q oscillator phase in degrees at the first point acquired.
 QSource	Gets or sets the channel that is the input to the Q data stream of the digital downconverter (DDC).

[Top](#)

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcCenterFrequency Property

Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 10 MHz. The valid value range is 0 to $(0.5 \times \text{SampleClockTimebaseRate})$ for digitizer).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, an exception is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcCenterFrequency Property

Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 10 MHz. The valid value range is 0 to $(0.5 \times \text{SampleClockTimebaseRate})$ for digitizer).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, an exception is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcEnabled Property

Gets or sets whether the digital down converter (DDC) block of the digitizer is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

if you need to use any of the onboard signal processing (OSP) functionality of your OSP-enabled instrument, or , to disable all other DDC-related properties and have no effect on the acquired signal.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationEnabled Prop

Gets or sets whether the frequency translation of the data around the selected center frequency down to baseband is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FrequencyTranslationEnabled { get; set; }
```

```
Public Property FrequencyTranslationEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationPhaseI Property

Gets or sets the I oscillator phase in degrees at the first point acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double FrequencyTranslationPhaseI { get; set; }
```

```
Public Property FrequencyTranslationPhaseI As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0. Valid values are -360 to 360, inclusive.

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationPhaseQ Prop

Gets or sets the Q oscillator phase in degrees at the first point acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double FrequencyTranslationPhaseQ { get; set; }
```

```
Public Property FrequencyTranslationPhaseQ As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 90. Valid values are -360 to 360, inclusive.

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcQSource Property

Gets or sets the channel that is the input to the Q data stream of the digital downconverter (DDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string QSource { get; set; }
```

```
Public Property QSource As String  
    Get  
    Set
```

Property Value

Type: [String](#)

The default value is the channel that the property is registered to. All valid channels for the instrument are valid values.

Remarks

For multi-instrument sessions, the channel you define with this property as the Q data stream and the channel for the corresponding I data stream must both be located within the same physical instrument.

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the PXI/PCI-5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdc Properties

The [ScopeChannelDdc](#) type exposes the following members.

Properties

Name	Description
 CenterFrequency	Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.
 Enabled	Gets or sets whether the digital down converter (DDC) block of the digitizer is enabled.
 FrequencyTranslationEnabled	Gets or sets whether the frequency translation of the data around the selected center frequency down to baseband is enabled.
 FrequencyTranslationPhaseI	Gets or sets the I oscillator phase in degrees at the first point acquired.
 FrequencyTranslationPhaseQ	Gets or sets the Q oscillator phase in degrees at the first point acquired.
 QSource	Gets or sets the channel that is the input to the Q data stream of the digital downconverter (DDC).

[Top](#)

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcCenterFrequency Property

Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10 MHz. The valid value range is 0 to $(0.5 \times \text{SampleClockTimebaseRate})$ for digitizer).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, an exception is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcCenterFrequency Property

Gets or sets the frequency, in hertz, at which the DDC block frequency translates the input data. This frequency is the region of the spectrum that you want to downconvert to baseband.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 10 MHz. The valid value range is 0 to $(0.5 \times \text{SampleClockTimebaseRate})$ for digitizer).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, an exception is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcEnabled Property

Gets or sets whether the digital down converter (DDC) block of the digitizer is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

if you need to use any of the onboard signal processing (OSP) functionality of your OSP-enabled instrument, or , to disable all other DDC-related properties and have no effect on the acquired signal.

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationEnabled Property

Gets or sets whether the frequency translation of the data around the selected center frequency down to baseband is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FrequencyTranslationEnabled { get; set; }
```

```
Public Property FrequencyTranslationEnabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationPhaseI Property

Gets or sets the I oscillator phase in degrees at the first point acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double FrequencyTranslationPhaseI { get; set; }
```

```
Public Property FrequencyTranslationPhaseI As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0. Valid values are -360 to 360, inclusive.

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcFrequencyTranslationPhaseQ Property

Gets or sets the Q oscillator phase in degrees at the first point acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double FrequencyTranslationPhaseQ { get; set; }
```

```
Public Property FrequencyTranslationPhaseQ As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 90. Valid values are -360 to 360, inclusive.

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdcQSource Property

Gets or sets the channel that is the input to the Q data stream of the digital downconverter (DDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string QSource { get; set; }
```

```
Public Property QSource As String  
    Get  
    Set
```

Property Value

Type: [String](#)

The default value is the channel that the property is registered to. All valid channels for the instrument are valid values.

Remarks

For multi-instrument sessions, the channel you define with this property as the Q data stream and the channel for the corresponding I data stream must both be located within the same physical instrument.

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the PXI/PCI-5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelDdc Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelDdc Methods

The [ScopeChannelDdc](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelDdc Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilter Class

Provides configuration options for the equalization filter. The equalization filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeChannelEqualizationFilter

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public sealed class ScopeChannelEqualizationFilter : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelEqualizationFilter
    Inherits ScopeSubObject
```


The ScopeChannelEqualizationFilter type exposes the following members.






Properties

	Name	Description
	<u>Enabled</u>	Gets or sets whether the onboard signal processing equalization FIR block is enabled.
	<u>NumberOfCoefficients</u>	Gets the number of coefficients that the equalization FIR filter can accept.

Top

Methods

	Name	Description
	<u>ConfigureCoefficients</u>	Configures the custom coefficients for the equalization FIR filter on the instrument.

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetCoefficients	Gets the custom coefficients for the equalization FIR filter on the instrument.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

Ensure that you call [ConfigureCoefficients\(Double\)](#) with a double array whose size matches [NumberOfCoefficients](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilter Properties

The [ScopeChannelEqualizationFilter](#) type exposes the following members.

Properties

Name	Description
------	-------------

 Enabled	Gets or sets whether the onboard signal processing equalization FIR block is enabled.
 NumberOfCoefficients	Gets the number of coefficients that the equalization FIR filter can accept.

[Top](#)

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterEnabled Property

Gets or sets whether the onboard signal processing equalization FIR block is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

If you set Enabled to `true`, ensure that you call [ConfigureCoefficients\(Double\)](#) with a double array whose size matches [NumberOfCoefficients](#).

Note

This property is valid only with high-speed digitizers that support onboard signal processing, such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterEnabled Property

Gets or sets whether the onboard signal processing equalization FIR block is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

If you set Enabled to , ensure that you call [ConfigureCoefficients\(Double\)](#) with a double array whose size matches [NumberOfCoefficients](#).

Note

This property is valid only with high-speed digitizers that support onboard signal processing, such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterNumberOfCoefficients

Gets the number of coefficients that the equalization FIR filter can accept.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public int NumberOfCoefficients { get; }
```

```
Public ReadOnly Property NumberOfCoefficients As Integer  
    Get
```

Property Value

Type: [Int32](#)

The number of coefficients that the equalization FIR filter can accept.

Remarks

Use this property to get the size of the double array to pass to [ConfigureCoefficients\(Double\)](#).

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilter Properties

The [ScopeChannelEqualizationFilter](#) type exposes the following members.

Properties

	Name	Description
	Enabled	Gets or sets whether the onboard signal processing equalization FIR block is enabled.
	NumberOfCoefficients	Gets the number of coefficients that the equalization FIR filter can accept.

[Top](#)

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterEnabled Property

Gets or sets whether the onboard signal processing equalization FIR block is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

If you set Enabled to , ensure that you call [ConfigureCoefficients\(Double\)](#) with a double array whose size matches [NumberOfCoefficients](#).

 Note

This property is valid only with high-speed digitizers that support onboard signal processing, such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterEnabled Property

Gets or sets whether the onboard signal processing equalization FIR block is enabled.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

If you set Enabled to `true`, ensure that you call [ConfigureCoefficients\(Double\)](#) with a double array whose size matches [NumberOfCoefficients](#).

Note

This property is valid only with high-speed digitizers that support onboard signal processing, such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterNumberOfCoefficients Property

Gets the number of coefficients that the equalization FIR filter can accept.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int NumberOfCoefficients { get; }
```

```
Public ReadOnly Property NumberOfCoefficients As Integer
    Get
```

Property Value

Type: [Int32](#)

The number of coefficients that the equalization FIR filter can accept.

Remarks

Use this property to get the size of the double array to pass to [ConfigureCoefficients\(Double\)](#).

Note

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilter Methods

The [ScopeChannelEqualizationFilter](#) type exposes the following members.

Methods

Name	Description
ConfigureCoefficients	Configures the custom coefficients for the equalization FIR filter on the instrument.
Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
GetCoefficients	Gets the custom coefficients for the equalization FIR filter on the instrument.
GetHashCode	Serves as the default hash function. (Inherited from Object .)
GetType	Gets the Type of the current instance. (Inherited from Object .)
ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterConfigureCoefficients

Configures the custom coefficients for the equalization FIR filter on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ConfigureCoefficients(  
    double[] coefficients  
)
```

```
Public Sub ConfigureCoefficients (  
    coefficients As Double()  
)
```

Parameters

coefficients

Type: [SystemDouble](#)

Pass the custom coefficients for the equalization FIR filter on the instrument.

These coefficients should be between +1 and -1. The length of the array is

[NumberOfCoefficients](#).

Remarks

If the equalization filter is [Enabled](#), ensure you call this method with a double array whose size matches [NumberOfCoefficients](#). Because this filter is a generic FIR filter, any coefficients within the range specified are valid.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterConfigureCoefficients Method

Configures the custom coefficients for the equalization FIR filter on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ConfigureCoefficients(  
    double[] coefficients  
)
```

```
Public Sub ConfigureCoefficients (  
    coefficients As Double()  
)
```

Parameters

coefficients

Type: [SystemDouble](#)

Pass the custom coefficients for the equalization FIR filter on the instrument.

These coefficients should be between +1 and -1. The length of the array is [NumberOfCoefficients](#).

Remarks

If the equalization filter is [Enabled](#), ensure you call this method with a double array whose size matches [NumberOfCoefficients](#). Because this filter is a generic FIR filter, any coefficients within the range specified are valid.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelEqualizationFilterGetCoefficients Method

Gets the custom coefficients for the equalization FIR filter on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double[] GetCoefficients()
```

```
Public Function GetCoefficients As Double()
```

Return Value

Type: [Double](#)

The custom coefficients for the equalization FIR filter on the instrument.

Exceptions

Exception	Condition
ObjectDisposedException	GetCoefficients was called after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

This filter is designed to compensate the input signal for artifacts introduced to the signal outside of the digitizer. Because this filter is a generic FIR filter any coefficients are valid.

Coefficient values should be between +1 and -1.

See Also

Reference

[ScopeChannelEqualizationFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Class

Represents the ScopeChannelMeasurement sub-objects that contain properties to allow you to configure the digitizers to obtain scalar and array measurements.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurement

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax










C#VB




```
public sealed class ScopeChannelMeasurement : ScopeSubObject,
    ISupportSynchronizationContext, IIVI_ScopeChannelMeasurement
```

```
Public NotInheritable Class ScopeChannelMeasurement
    Inherits ScopeSubObject
    Implements ISupportSynchronizationContext, IIVI_ScopeChannelMeasurement
```

The ScopeChannelMeasurement type exposes the following members.







Properties







Name	Description
 ArrayGain	Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.
 ArrayOffset	Gets or sets a scalar value that is added to every element of an array during the Array Offset measurement.
 Filter	Gets measurement properties related to filtering.
 HysteresisPercent	Gets or sets the percentage of the full-scale vertical range for the hysteresis window size. Digital hysteresis is used in several scalar waveform measurements.
 InterpolationSamplingFactor	Gets or sets, indirectly, the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.
 LastHistogramSize	Gets or sets the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.
 OtherChannel	Gets or sets the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.
 PolynomialInterpolationOrder	Gets or sets the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 specifies linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.
 ReferenceLevel	Gets properties related to different reference levels.

 SynchronizeCallbacks	Specifies how events and callback delegates are invoked.
 TimeHistogram	Gets properties that are related to time histogram measurements.
 VoltageHistogram	Gets properties that are related to voltage histogram measurements.









[Top](#)








Methods






Name	Description
 AddWaveformProcessing	Adds one measurement to the list of processing steps that are completed before the measurement.
 BeginFetchArrayMeasurement	Asynchronously obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 BeginFetchByte	Begins an asynchronous retrieval of data from a previously initiated acquisition.
 BeginFetchComplexDouble	Begins an asynchronous retrieval of data that the digitizer has acquired from a previously initiated acquisition.
 BeginFetchComplexInt16	Begins an asynchronous retrieval of data from single channels and records.
 BeginFetchDouble	Asynchronously returns the waveform from a previously initiated acquisition that the






	digitizer acquires for the specified channel. Refer to Using Fetch Functions for more information on using this method.
 BeginFetchInt16	Begins an asynchronous retrieval of data from a previously initiated acquisition.
 BeginFetchInt32	Begins an asynchronous retrieval of data from a previously initiated acquisition. Refer to Using Fetch Functions for more information on using this method.
 BeginFetchScalarMeasurement	Begins an asynchronous fetching of a waveform from the digitizer and performs the specified waveform measurement. Refer to Using Fetch Functions for more information.
 BeginFetchScalarMeasurementStatistics	Asynchronously obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.
 BeginRead	Asynchronously initiates an acquisition, waits for it to complete, and retrieves the data.
 BeginReadScalarMeasurement	Asynchronously initiates an acquisition, waits for it to complete, and performs the

	specified waveform measurement for a single channel and record or for multiple channels and records. Refer to Using Fetch Functions for more information.
ClearWaveformMeasurements	Clears the waveform statistics on the channel and for all measurements.
ClearWaveformMeasurements(ScopeArrayMeasurementType)	Clears the waveform statistics on the channel and for scalar measurements.
ClearWaveformMeasurements(ScopeScalarMeasurementType)	Clears the waveform statistics on the channel and for scalar measurements.
ClearWaveformProcessing	Clears the list of processing steps assigned to the given channel.
EndFetchArrayMeasurement(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
EndFetchArrayMeasurement(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.
EndFetchByte(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
EndFetchByte(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.
EndFetchComplexDouble(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
EndFetchComplexDouble(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.






 <u>EndFetchComplexInt16(IAsyncResult)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchComplexInt16(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchDouble(IAsyncResult)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchDouble(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchInt16(IAsyncResult)</u>	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndFetchInt16(IAsyncResult, ScopeWaveformInfo)</u>	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndFetchInt32(IAsyncResult)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchInt32(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous fetch to complete.

 <u>EndFetchScalarMeasurement</u>	<p>Waits for a pending asynchronous fetch to complete.</p>
 <u>EndFetchScalarMeasurementStatistics</u>	<p>Waits for a pending asynchronous fetch to complete.</p>
 <u>EndRead(IAsyncResult)</u>	<p>Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
 <u>EndRead(IAsyncResult, ScopeWaveformInfo)</u>	<p>Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
 <u>EndReadScalarMeasurement</u>	<p>Waits for a pending asynchronous read to complete.</p>
 <u>Equals</u>	<p>Determines whether the specified object is equal to the current object. (Inherited from <u>Object</u>.)</p>
 <u>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble)</u>	<p>Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>

<p> <u>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</u></p>	<p>Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
<p> <u>FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
<p> <u>FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte, ScopeWaveformInfo)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify. Refer to <u>Features Supported by Device</u> for more information on using this method.</p>
<p> <u>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)</u></p>	<p>Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.</p>
<p> <u>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble, ScopeWaveformInfo)</u></p>	<p>Retrieves data that the digitizer has acquired from a previously initiated</p>

	acquisition and returns a one-dimensional array of complex, scaled waveforms.
 <u>FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16)</u>	Retrieves data from single channels and records. A one-dimensional array of complex binary 16-bit waveforms.
 <u>FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16, ScopeWaveformInfo)</u>	Retrieves data from single channels and records. Returns a one-dimensional array of complex binary 16-bit waveforms.
 <u>FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)</u>	Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to <u>Using Fetch Functions</u> for more information on using this method.
 <u>FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</u>	Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to <u>Using Fetch Functions</u> for more information on using this method.
 <u>FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16)</u>	Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition

		type, and the number of records you specify.
⇒	<u>FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16, ScopeWaveformInfo)</u>	Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
⇒	<u>FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32)</u>	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
⇒	<u>FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32, ScopeWaveformInfo)</u>	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
⇒	<u>FetchScalarMeasurement</u>	Fetches a waveform from the digitizer and performs the specified waveform measurement. Refer to <u>Using Fetch Functions</u> for more information.
⇒	<u>FetchScalarMeasurementStatistics</u>	Obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results

	depending on the number of channels, the acquisition type, and the number of records you specify.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 ReadScalarMeasurement	Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records. Refer to Using Fetch Functions for more information.
 ToString	Returns a string that represents the current object.

(Inherited from [Object](#).)[Top](#)

Remarks

For more information, refer to Making Waveform Measurements in the [NI High-Speed Digitizers Help](#) or to the Scalar Measurements and Array Measurements topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources




[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)










Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Properties

The [ScopeChannelMeasurement](#) type exposes the following members.

Properties

Name	Description
 ArrayGain	Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.
 ArrayOffset	Gets or sets a scalar value that is added to every element of an array during the Array Offset measurement.
 Filter	Gets measurement properties related to filtering.

 HysteresisPercent	Gets or sets the percentage of the full-scale vertical range for the hysteresis window size. Digital hysteresis is used in several scalar waveform measurements.
 InterpolationSamplingFactor	Gets or sets, indirectly, the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.
 LastHistogramSize	Gets or sets the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.
 OtherChannel	Gets or sets the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.
 PolynomialInterpolationOrder	Gets or sets the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 specifies linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.
 ReferenceLevel	Gets properties related to different reference levels.
 SynchronizeCallbacks	Specifies how events and callback delegates are invoked.
 TimeHistogram	Gets properties that are related to time histogram measurements.
 VoltageHistogram	Gets properties that are related to voltage histogram measurements.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayGain Property

Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayGain { get; set; }
```

```
Public Property ArrayGain As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayGain Property

Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayGain { get; set; }
```

```
Public Property ArrayGain As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayOffset Property

Gets or sets a scalar value that is added to every element of an array during the Array Offset measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayOffset { get; set; }
```

```
Public Property ArrayOffset As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Property

Gets measurement properties related to filtering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementFilter Filter { get; }
```

```
Public ReadOnly Property Filter As ScopeChannelMeasurementFilter  
    Get
```

Property Value

Type: [ScopeChannelMeasurementFilter](#)

An object of type [ScopeChannelMeasurementFilter](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementHysteresisPercent Property

Gets or sets the percentage of the full-scale vertical range for the hysteresis window size. Digital hysteresis is used in several scalar waveform measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HysteresisPercent { get; set; }
```

```
Public Property HysteresisPercent As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 2%.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementInterpolationSamplingFac

Gets or sets, indirectly, the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double InterpolationSamplingFactor { get; set; }
```

```
Public Property InterpolationSamplingFactor As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 2.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementLastHistogramSize Property

Gets or sets the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public long LastHistogramSize { get; set; }
```

```
Public Property LastHistogramSize As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementOtherChannel Property

Gets or sets the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string OtherChannel { get; set; }
```

```
Public Property OtherChannel As String
    Get
    Set
```

Property Value

Type: [String](#)

The default value is 0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementPolynomialInterpolationOrder

Gets or sets the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 specifies linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long PolynomialInterpolationOrder { get; set; }
```

```
Public Property PolynomialInterpolationOrder As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 1.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Property

Gets properties related to different reference levels.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementReferenceLevel ReferenceLevel { get; }
```

```
Public ReadOnly Property ReferenceLevel As ScopeChannelMeasurementReferenceLevel
    Get
```


Property Value

Type: [ScopeChannelMeasurementReferenceLevel](#)

An object of type [ScopeChannelMeasurementReferenceLevel](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementSynchronizeCallbacks Property

Specifies how events and callback delegates are invoked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool SynchronizeCallbacks { get; set; }
```

```
Public Property SynchronizeCallbacks As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if the callbacks are synchronized; otherwise, . The default value is .

Implements

[ISupportSynchronizationContextSynchronizeCallbacks](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Property

Gets properties that are related to time histogram measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementTimeHistogram TimeHistogram { get; }
```

```
Public ReadOnly Property TimeHistogram As ScopeChannelMeasurementTimeHistogram  
    Get
```

Property Value

Type: [ScopeChannelMeasurementTimeHistogram](#)

An object of type [ScopeChannelMeasurementTimeHistogram](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Property

Gets properties that are related to voltage histogram measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public ScopeChannelMeasurementVoltageHistogram VoltageHistogram { get; }
```

```
Public ReadOnly Property VoltageHistogram As  
ScopeChannelMeasurementVoltageHistogram  
Get
```

Property Value

Type: [ScopeChannelMeasurementVoltageHistogram](#)

An object of type [ScopeChannelMeasurementVoltageHistogram](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)













[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Properties

The [ScopeChannelMeasurement](#) type exposes the following members.

Properties

Name	Description
 ArrayGain	Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.
 ArrayOffset	Gets or sets a scalar value that is added to every element of an array during the Array Offset measurement.
 Filter	Gets measurement properties related to filtering.
 HysteresisPercent	Gets or sets the percentage of the full-scale vertical range for the hysteresis window size. Digital hysteresis is used in several scalar waveform measurements.
 InterpolationSamplingFactor	Gets or sets, indirectly, the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.
 LastHistogramSize	Gets or sets the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.
 OtherChannel	Gets or sets the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.
 PolynomialInterpolationOrder	Gets or sets the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 specifies linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.
 ReferenceLevel	Gets properties related to different reference levels.
 SynchronizeCallbacks	Specifies how events and callback delegates are invoked.
 TimeHistogram	Gets properties that are related to time histogram measurements.
 VoltageHistogram	Gets properties that are related to voltage histogram measurements.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayGain Property

Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayGain { get; set; }
```

```
Public Property ArrayGain As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayGain Property

Gets or sets a scalar value that is multiplied by every element of an array during the Array Gain measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayGain { get; set; }
```

```
Public Property ArrayGain As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementArrayOffset Property

Gets or sets a scalar value that is added to every element of an array during the Array Offset measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ArrayOffset { get; set; }
```

```
Public Property ArrayOffset As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Property

Gets measurement properties related to filtering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementFilter Filter { get; }
```

```
Public ReadOnly Property Filter As ScopeChannelMeasurementFilter  
    Get
```

Property Value

Type: [ScopeChannelMeasurementFilter](#)

An object of type [ScopeChannelMeasurementFilter](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementHysteresisPercent Property

Gets or sets the percentage of the full-scale vertical range for the hysteresis window size. Digital hysteresis is used in several scalar waveform measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HysteresisPercent { get; set; }
```

```
Public Property HysteresisPercent As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 2%.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementInterpolationSamplingFactor Property

Gets or sets, indirectly, the new number of points for polynomial interpolation measurements. The new number of points is the input number of points times the sampling factor.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double InterpolationSamplingFactor { get; set; }
```

```
Public Property InterpolationSamplingFactor As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 2.0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementLastHistogramSize Property

Gets or sets the number of bins in the last acquisition histogram. This histogram is used to determine several scalar measurements, such as voltage low and voltage high.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long LastHistogramSize { get; set; }
```

```
Public Property LastHistogramSize As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementOtherChannel Property

Gets or sets the second channel for two-channel measurements. If processing steps are registered with this channel, the processing is done before the waveform is used in a two-channel measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string OtherChannel { get; set; }
```

```
Public Property OtherChannel As String
    Get
    Set
```

Property Value

Type: [String](#)

The default value is 0.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementPolynomialInterpolationOrder Property

Gets or sets the polynomial order used for the polynomial interpolation measurement. For example, an order of 1 specifies linear interpolation whereas an order of 2 specifies parabolic interpolation. Any positive integer is valid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long PolynomialInterpolationOrder { get; set; }
```

```
Public Property PolynomialInterpolationOrder As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 1.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Property

Gets properties related to different reference levels.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementReferenceLevel ReferenceLevel { get; }
```

```
Public ReadOnly Property ReferenceLevel As ScopeChannelMeasurementReferenceLevel
    Get
```

Property Value

Type: [ScopeChannelMeasurementReferenceLevel](#)

An object of type [ScopeChannelMeasurementReferenceLevel](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementSynchronizeCallbacks Property

Specifies how events and callback delegates are invoked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool SynchronizeCallbacks { get; set; }
```

```
Public Property SynchronizeCallbacks As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if the callbacks are synchronized; otherwise, . The default value is .

Implements

ISupportSynchronizationContextSynchronizeCallbacks

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Property

Gets properties that are related to time histogram measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementTimeHistogram TimeHistogram { get; }
```

```
Public ReadOnly Property TimeHistogram As ScopeChannelMeasurementTimeHistogram  
    Get
```

Property Value

Type: [ScopeChannelMeasurementTimeHistogram](#)

An object of type [ScopeChannelMeasurementTimeHistogram](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Property

Gets properties that are related to voltage histogram measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelMeasurementVoltageHistogram VoltageHistogram { get; }
```

```
Public ReadOnly Property VoltageHistogram As  
ScopeChannelMeasurementVoltageHistogram  
Get
```

Property Value

Type: [ScopeChannelMeasurementVoltageHistogram](#)

An object of type [ScopeChannelMeasurementVoltageHistogram](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurement Methods










The [ScopeChannelMeasurement](#) type exposes the following members.






Methods



Name	Description
 AddWaveformProcessing	Adds one measurement to the list of processing steps that are completed before the measurement.
 BeginFetchArrayMeasurement	Asynchronously obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 BeginFetchByte	Begins an asynchronous retrieval of data from a previously initiated acquisition.
 BeginFetchComplexDouble	Begins an asynchronous retrieval of data that the digitizer has acquired from a previously initiated acquisition.
 BeginFetchComplexInt16	Begins an asynchronous retrieval of data from single channels and records.
 BeginFetchDouble	Asynchronously returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. Refer to Using Fetch Functions for more information on using this method.
 BeginFetchInt16	Begins an asynchronous retrieval of data from a

	previously initiated acquisition.
 BeginFetchInt32	Begins an asynchronous retrieval of data from a previously initiated acquisition. Refer to Using Fetch Functions for more information on using this method.
 BeginFetchScalarMeasurement	Begins an asynchronous fetching of a waveform from the digitizer and performs the specified waveform measurement. Refer to Using Fetch Functions for more information.
 BeginFetchScalarMeasurementStatistics	Asynchronously obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.
 BeginRead	Asynchronously initiates an acquisition, waits for it to complete, and retrieves the data.
 BeginReadScalarMeasurement	Asynchronously initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records. Refer to Using Fetch Functions for more information.






<u>⇒ <code>ClearWaveformMeasurements</code></u>	Clears the waveform statistics on the channel and for all measurements.
<u>⇒ <code>ClearWaveformMeasurements(ScopeArrayMeasurementType)</code></u>	Clears the waveform statistics on the channel and for scalar measurements.
<u>⇒ <code>ClearWaveformMeasurements(ScopeScalarMeasurementType)</code></u>	Clears the waveform statistics on the channel and for scalar measurements.
<u>⇒ <code>ClearWaveformProcessing</code></u>	Clears the list of processing steps assigned to the given channel.
<u>⇒ <code>EndFetchArrayMeasurement(IAsyncResult)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchArrayMeasurement(IAsyncResult, ScopeWaveformInfo)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchByte(IAsyncResult)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchByte(IAsyncResult, ScopeWaveformInfo)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchComplexDouble(IAsyncResult)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchComplexDouble(IAsyncResult, ScopeWaveformInfo)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchComplexInt16(IAsyncResult)</code></u>	Waits for a pending asynchronous fetch to complete.
<u>⇒ <code>EndFetchComplexInt16(IAsyncResult, ScopeWaveformInfo)</code></u>	Waits for a pending asynchronous fetch to complete.

 <u>EndFetchDouble(IAsyncResult)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchDouble(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchInt16(IAsyncResult)</u>	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndFetchInt16(IAsyncResult, ScopeWaveformInfo)</u>	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndFetchInt32(IAsyncResult)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchInt32(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchScalarMeasurement</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndFetchScalarMeasurementStatistics</u>	Waits for a pending asynchronous fetch to complete.
 <u>EndRead(IAsyncResult)</u>	Waits for a pending asynchronous read to

	complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndRead(IAsyncResult, ScopeWaveformInfo)</u>	Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>EndReadScalarMeasurement</u>	Waits for a pending asynchronous read to complete.
 <u>Equals</u>	Determines whether the specified object is equal to the current object. (Inherited from <u>Object.</u>)
 <u>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble)</u>	Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 <u>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</u>	Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

<p> <u>FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
<p> <u>FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte, ScopeWaveformInfo)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify. Refer to <u>Features Supported by Device</u> for more information on using this method.</p>
<p> <u>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)</u></p>	<p>Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.</p>
<p> <u>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble, ScopeWaveformInfo)</u></p>	<p>Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.</p>
<p> <u>FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16)</u></p>	<p>Retrieves data from single channels and records. A one-dimensional array of complex binary 16-bit waveforms.</p>

<p>⇒ <u>FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16, ScopeWaveformInfo)</u></p>	<p>Retrieves data from single channels and records. Returns a one-dimensional array of complex binary 16-bit waveforms.</p>
<p>⇒ <u>FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)</u></p>	<p>Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to <u>Using Fetch Functions</u> for more information on using this method.</p>
<p>⇒ <u>FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</u></p>	<p>Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to <u>Using Fetch Functions</u> for more information on using this method.</p>
<p>⇒ <u>FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.</p>
<p>⇒ <u>FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16, ScopeWaveformInfo)</u></p>	<p>Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms</p>

	depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32)	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32, ScopeWaveformInfo)	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchScalarMeasurement	Fetches a waveform from the digitizer and performs the specified waveform measurement. Refer to Using Fetch Functions for more information.
 FetchScalarMeasurementStatistics	Obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)

☰	GetType	Gets the Type of the current instance. (Inherited from Object .)
☰	Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
☰	Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
☰	ReadScalarMeasurement	Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records. Refer to Using Fetch Functions for more information.
☰	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)[See Also](#)[Reference](#)[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementAddWaveformProcessing

Adds one measurement to the list of processing steps that are completed before the measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void AddWaveformProcessing(  
    ScopeArrayMeasurementType arrayMeasurementType  
)
```

```
Public Sub AddWaveformProcessing (  
    arrayMeasurementType As ScopeArrayMeasurementType  
)
```

Parameters

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)
The array measurement to add.

Remarks

The processing is added on a per-channel basis, and the processing measurements are completed in the same order in which they are registered. The channel list considered is the channel associated with this measurement object. All measurement library properties (properties present in [ScopeChannelMeasurement](#) and its sub-objects) are cached at the time of registering the processing, and this set

of parameters is used during the processing step. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing is done before any other measurements.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementAddWaveformProcessing Method

Adds one measurement to the list of processing steps that are completed before the measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void AddWaveformProcessing(  
    ScopeArrayMeasurementType arrayMeasurementType  
)
```

```
Public Sub AddWaveformProcessing (  
    arrayMeasurementType As ScopeArrayMeasurementType  
)
```

Parameters

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The array measurement to add.

Remarks

The processing is added on a per-channel basis, and the processing measurements are completed in the same order in which they are registered. The channel list considered is the channel associated with this measurement object. All measurement library properties (properties present in [ScopeChannelMeasurement](#) and its sub-objects) are cached at the time of registering the processing, and this set of parameters is used during the processing step. The processing measurements are streamed, so the result of the first processing step is used as the input for the next step. The processing is done before any other measurements.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchArrayMeasurement Method

Asynchronously obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginFetchArrayMeasurement(
    PrecisionTimeSpan timeout,
    ScopeArrayMeasurementType arrayMeasurementType,
    AnalogWaveformCollection<double> waveforms,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchArrayMeasurement (
    timeout As PrecisionTimeSpan,
    arrayMeasurementType As ScopeArrayMeasurementType,
    waveforms As AnalogWaveformCollection(Of Double),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The type of the array measurement to perform.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionDouble`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to `.` The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output

waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchByte Method

Begins an asynchronous retrieval of data from a previously initiated acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)
[VB](#)

```
public IAsyncResult BeginFetchByte(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<byte> waveforms,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchByte (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Byte),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever data is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollection`[Byte](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform

object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchComplexDouble Method

Begins an asynchronous retrieval of data that the digitizer has acquired from a previously initiated acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginFetchComplexDouble(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexDouble> waveforms,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchComplexDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexDouble),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all

available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstruments.ComplexWaveformCollection.ComplexDouble`
An array of type `ComplexWaveformCollection.TData`.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchComplexInt16 Method

Begins an asynchronous retrieval of data from single channels and records.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

[C#](#)
[VB](#)

```
public IAsyncResult BeginFetchComplexInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexInt16> waveforms,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchComplexInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexInt16),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexInt16`

An array of type `ComplexWaveformCollectionTData`.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchDouble Method

Asynchronously returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. Refer to [Using Fetch Functions](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public IAsyncResult BeginFetchDouble(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,
```

```

    AnalogWaveformCollection<double> waveforms,
    AsyncCallback callback,
    Object asyncState
)

```

```

Public Function BeginFetchDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Double),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult

```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

A scaled voltage waveform array of type [AnalogWaveformCollectionTData](#).

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchInt16 Method

Begins an asynchronous retrieval of data from a previously initiated acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public IAsyncResult BeginFetchInt16(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<short> waveforms,  
    AsyncCallback callback,
```

```

        Object asyncState
    )

```

```

Public Function BeginFetchInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Short),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult

```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use `-1` for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionInt16](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `Nothing`. The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output

waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchInt32 Method

Begins an asynchronous retrieval of data from a previously initiated acquisition.

Refer to [Using Fetch Functions](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginFetchInt32(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<int> waveforms,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchInt32 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Integer),
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionInt32`

The waveform object whose memory can be reused while creating the output

waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchScalarMeasurement Method

Begins an asynchronous fetching of a waveform from the digitizer and performs the specified waveform measurement. Refer to [Using Fetch Functions](#) for more information.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginFetchScalarMeasurement(
    PrecisionTimeSpan timeout,
    ScopeScalarMeasurementType scalarMeasurementType,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginFetchScalarMeasurement (
    timeout As PrecisionTimeSpan,
    scalarMeasurementType As ScopeScalarMeasurementType,
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The type of the scalar measurement to be performed.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginFetchScalarMeasurementStatistics
Method

Asynchronously obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public IAsyncResult BeginFetchScalarMeasurementStatistics(
    PrecisionTimeSpan timeout,
```

```

        ScopeScalarMeasurementType scalarMeasurementType,
        AsyncCallback callback,
        Object asyncState
    )

```

```

Public Function BeginFetchScalarMeasurementStatistics (
    timeout As PrecisionTimeSpan,
    scalarMeasurementType As ScopeScalarMeasurementType,
    callback As AsyncCallback,
    asyncState As Object
) As IAsyncResult

```

Parameters

timeout

Type: [NationalInstruments.PrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The type of the scalar measurement to be performed on each fetched waveform.

callback

Type: [SystemAsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [SystemObject](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginRead Method

Asynchronously initiates an acquisition, waits for it to complete, and retrieves the data.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginRead(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<double> waveforms,  
    AsyncCallback callback,  
    Object asyncState  
)
```

```
Public Function BeginRead (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Double),  
    callback As AsyncCallback,  
    asyncState As Object  
) As IAsyncResult
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionDouble`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

callback

Type: `SystemAsyncCallback`

A delegate of type `AsyncCallback`, which will be invoked once the operation is complete.

asyncState

Type: `SystemObject`

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous read, which could still be pending.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementBeginReadScalarMeasurement Method

Asynchronously initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records. Refer to [Using Fetch Functions](#) for more information.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public IAsyncResult BeginReadScalarMeasurement (
    PrecisionTimeSpan timeout,
    ScopeScalarMeasurementType scalarMeasurementType,
    AsyncCallback callback,
    Object asyncState
)
```

```
Public Function BeginReadScalarMeasurement (
    timeout As PrecisionTimeSpan,
    scalarMeasurementType As ScopeScalarMeasurementType,
    callback As AsyncCallback,
```



```

        asyncState As Object
    ) As IAsyncResult

```

Parameters

timeout

Type: [NationalInstruments.PrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The type of the scalar measurement to perform.

callback

Type: [System.AsyncCallback](#)

A delegate of type [AsyncCallback](#), which will be invoked once the operation is complete.

asyncState

Type: [System.Object](#)

The state of the asynchronous operation.

Return Value

Type: [IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous read.

See Also

Reference




[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformMeasurements Method

Overload List

Name	Description
 ClearWaveformMeasurements	Clears the waveform statistics on the channel and for all measurements.
 ClearWaveformMeasurements(ScopeArrayMeasurementType)	Clears the waveform statistics on the channel and for scalar measurements.
 ClearWaveformMeasurements(ScopeScalarMeasurementType)	Clears the waveform statistics on the channel and for scalar measurements.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformMeasurements

Clears the waveform statistics on the channel and for all measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ClearWaveformMeasurements()
```

```
Public Sub ClearWaveformMeasurements
```

Remarks

Every time a measurement is called, the statistics information is updated, including the min, max, mean, standard deviation, and number of updates.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[ClearWaveformMeasurements Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformMeasurements Method

Clears the waveform statistics on the channel and for all measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ClearWaveformMeasurements()
```

```
Public Sub ClearWaveformMeasurements
```

Remarks

Every time a measurement is called, the statistics information is updated, including the min, max, mean, standard deviation, and number of updates.

See Also

Reference

[ScopeChannelMeasurement Class](#)

ClearWaveformMeasurements Overload

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformMeasurements Method
(ScopeArrayMeasurementType)

Clears the waveform statistics on the channel and for scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ClearWaveformMeasurements (  
    ScopeArrayMeasurementType arrayMeasurementType  
)
```

```
Public Sub ClearWaveformMeasurements (  
    arrayMeasurementType As ScopeArrayMeasurementType  
)
```

Parameters

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The array measurement to clear the statistics for.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[ClearWaveformMeasurements Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformMeasurements Method (ScopeScalarMeasurementType)

Clears the waveform statistics on the channel and for scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ClearWaveformMeasurements (  
    ScopeScalarMeasurementType scalarMeasurementType  
)
```

```
Public Sub ClearWaveformMeasurements (  
    scalarMeasurementType As ScopeScalarMeasurementType  
)
```

Parameters

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The scalar measurement to clear the statistics for.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[ClearWaveformMeasurements Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementClearWaveformProcessing Method

Clears the list of processing steps assigned to the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ClearWaveformProcessing()
```

```
Public Sub ClearWaveformProcessing
```

Remarks

The processing steps are added with [AddWaveformProcessing\(ScopeArrayMeasurementType\)](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchArrayMeasurement Method

Overload List

Name	Description
------	-------------

 EndFetchArrayMeasurement(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
 EndFetchArrayMeasurement(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchArrayMeasurement

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public AnalogWaveformCollection<double> EndFetchArrayMeasurement (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchArrayMeasurement (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchArrayMeasurement Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchArrayMeasurement Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> EndFetchArrayMeasurement(  
    IAsyncResult asyncResult  
)
```



```
Public Function EndFetchArrayMeasurement (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchArrayMeasurement Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchArrayMeasurement Method
(IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public AnalogWaveformCollection<double> EndFetchArrayMeasurement (
    IAsyncResult asyncResult,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function EndFetchArrayMeasurement (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects containing the timing and scaling information.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)



[EndFetchArrayMeasurement Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchByte Method

Overload List

Name	Description
 EndFetchByte(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
 EndFetchByte(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchByte Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public AnalogWaveformCollection<byte> EndFetchByte (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchByte (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Byte)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [AnalogWaveformCollectionByte](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchByte Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<byte> EndFetchByte (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchByte (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Byte)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [AnalogWaveformCollectionByte](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchByte Method (IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<byte> EndFetchByte(  
    IAsyncResult asyncResult,  
    out ScopeWaveformInfo[] waveformInfo  
)
```

```
Public Function EndFetchByte (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Byte)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionByte](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)


[EndFetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexDouble Method

Overload List

Name	Description
 EndFetchComplexDouble(IAsyncResult)	Waits for a pending asynchronous fetch to complete.

 EndFetchComplexDouble(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.
--	---

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexDouble

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ComplexWaveformCollection<ComplexDouble> EndFetchComplexDouble (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchComplexDouble (
    asyncResult As IAsyncResult
) As ComplexWaveformCollection(Of ComplexDouble)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: `ComplexWaveformCollectionComplexDouble`
An array of type `ComplexWaveformCollectionTData`.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchComplexDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexDouble Method
(`IAsyncResult`)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in
`NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexDouble> EndFetchComplexDouble(  
    IAsyncResult asyncResult  
)
```

```
Public Function EndFetchComplexDouble (  
    asyncResult As IAsyncResult  
) As ComplexWaveformCollection(Of ComplexDouble)
```


Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [ComplexWaveformCollectionComplexDouble](#)

An array of type [ComplexWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchComplexDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexDouble Method
(IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexDouble> EndFetchComplexDouble (  
    IAsyncResult asyncResult,  
    out ScopeWaveformInfo[] waveformInfo  
)
```

```
Public Function EndFetchComplexDouble (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As ComplexWaveformCollection(Of ComplexDouble)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contain timing and scaling information about each waveform.

Return Value

Type: [ComplexWaveformCollectionComplexDouble](#)

An array of type [ComplexWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchComplexDouble Overload](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexInt16 Method

Overload List

Name	Description
 EndFetchComplexInt16(IAsyncResult)	Waits for a pending asynchronous fetch to complete.

 [EndFetchComplexInt16\(IAsyncResult, ScopeWaveformInfo\)](#)

Waits for a pending asynchronous fetch to complete.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexInt16 M

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ComplexWaveformCollection<ComplexInt16> EndFetchComplexInt16(
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchComplexInt16 (
    asyncResult As IAsyncResult
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: `ComplexWaveformCollectionComplexInt16`
 An array of type `ComplexWaveformCollectionTData`.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexInt16 Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexInt16> EndFetchComplexInt16(
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchComplexInt16 (
    asyncResult As IAsyncResult
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: `ComplexWaveformCollectionComplexInt16`
 An array of type `ComplexWaveformCollectionTData`.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchComplexInt16 Method (IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexInt16> EndFetchComplexInt16(
    IAsyncResult asyncResult,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function EndFetchComplexInt16 (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [ComplexWaveformCollectionComplexInt16](#)

An array of type [ComplexWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)



[EndFetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchDouble Method

Overload List

	Name	Description
	EndFetchDouble(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
	EndFetchDouble(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchDouble Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> EndFetchDouble(  
    IAsyncResult asyncResult  
)
```

```
Public Function EndFetchDouble (  
    asyncResult As IAsyncResult  
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchDouble Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> EndFetchDouble (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchDouble (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [SystemIAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchDouble Method (IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> EndFetchDouble(
    IAsyncResult asyncResult,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function EndFetchDouble (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [SystemIAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contain timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)



[EndFetchDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt16 Method

Overload List

Name	Description
 EndFetchInt16(IAsyncResult)	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 EndFetchInt16(IAsyncResult, ScopeWaveformInfo)	Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt16 Method (IAsyncResult)

Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public AnalogWaveformCollection<short> EndFetchInt16(  
    IAsyncResult asyncResult  
)
```

```
Public Function EndFetchInt16 (  
    asyncResult As IAsyncResult  
) As AnalogWaveformCollection(Of Short)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionInt16](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt16 Method (IAsyncResult)

Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<short> EndFetchInt16(
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchInt16 (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Short)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionInt16](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt16 Method (IAsyncResult, ScopeWaveformInfo)

Waits for pending asynchronous fetch to complete and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<short> EndFetchInt16(  
    IAsyncResult asyncResult,  
    out ScopeWaveformInfo[] waveformInfo  
)
```

```
Public Function EndFetchInt16 (  
    asyncResult As IAsyncResult,  
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()  
) As AnalogWaveformCollection(Of Short)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionInt16](#)

An array of type [AnalogWaveformCollectionTData](#) and the length of the array is the number of samples times the number of waveforms. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)



[EndFetchInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt32 Method

Overload List

	Name	Description
	EndFetchInt32(IAsyncResult)	Waits for a pending asynchronous fetch to complete.
	EndFetchInt32(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous fetch to complete.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt32 Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<int> EndFetchInt32(  
    IAsyncResult asyncResult  
)
```

```
Public Function EndFetchInt32 (  
    asyncResult As IAsyncResult  
) As AnalogWaveformCollection(Of Integer)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values: index 0 = record 0, channel 0 index x

= record 0, channel 1 index 2x = record 1, channel 0 index 3x = record 1, channel 1
Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt32 Method (IAsyncResult)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<int> EndFetchInt32 (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchInt32 (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Integer)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the asynchronous fetch, which could still be pending.

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values: index 0 = record 0, channel 0 index x = record 0, channel 1 index 2x = record 1, channel 0 index 3x = record 1, channel 1 Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchInt32 Method (IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<int> EndFetchInt32 (
    IAsyncResult asyncResult,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function EndFetchInt32 (
    asyncResult As IAsyncResult,
```

```
<OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()  
) As AnalogWaveformCollection(Of Integer)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contain timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values: index 0 = record 0, channel 0 index x = record 0, channel 1 index 2x = record 1, channel 0 index 3x = record 1, channel 1 Where x = the record length

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndFetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchScalarMeasurement Method

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in

[NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public double[] EndFetchScalarMeasurement (  
    IAsyncResult asyncResult  
)
```

```
Public Function EndFetchScalarMeasurement (  
    asyncResult As IAsyncResult  
) As Double()
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [Double](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndFetchScalarMeasurementStatistics Method

Waits for a pending asynchronous fetch to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public ScopeScalarMeasurementStatistics[] EndFetchScalarMeasurementStatistics (
    IAsyncResult asyncResult
)
```

```
Public Function EndFetchScalarMeasurementStatistics (
    asyncResult As IAsyncResult
) As ScopeScalarMeasurementStatistics()
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous fetch.

Return Value

Type: [ScopeScalarMeasurementStatistics](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference


[ScopeChannelMeasurement Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndRead Method

Overload List

Name	Description
 EndRead(IAsyncResult)	Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of

	channels, the acquisition type, and the number of records you specify.
 EndRead(IAsyncResult, ScopeWaveformInfo)	Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)[See Also](#)[Reference](#)[ScopeChannelMeasurement Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndRead Method (IAsyncResult)

Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

[Syntax](#)[C#VB](#)

```
public AnalogWaveformCollection<double> EndRead(
    IAsyncResult asyncResult
)
```

```
Public Function EndRead (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous read.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndRead Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndRead Method (IAsyncResult)

Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public AnalogWaveformCollection<double> EndRead(
    IAsyncResult asyncResult
)
```

```
Public Function EndRead (
    asyncResult As IAsyncResult
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous read.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndRead Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndRead Method (IAsyncResult, ScopeWaveformInfo)

Waits for a pending asynchronous read to complete. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> EndRead(
    IAsyncResult asyncResult,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function EndRead (
    asyncResult As IAsyncResult,
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Double)
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous read.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

See Also

Reference

[ScopeChannelMeasurement Class](#)

[EndRead Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementEndReadScalarMeasurement Method

Waits for a pending asynchronous read to complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double[] EndReadScalarMeasurement (
    IAsyncResult asyncResult
)
```

```
Public Function EndReadScalarMeasurement (
    asyncResult As IAsyncResult
) As Double()
```

Parameters

asyncResult

Type: [System.IAsyncResult](#)

An [IAsyncResult](#) that represents the pending asynchronous read.

Return Value

Type: [Double](#)

An array of type AnalogWaveformCollectionTData.

See Also

Reference



[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchArrayMeasurement Method

Overload List

Name	Description
 FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble)	Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble, ScopeWaveformInfo)	Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchArrayMeasurement)

Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchArrayMeasurement(
    PrecisionTimeSpan timeout,
    ScopeArrayMeasurementType arrayMeasurementType,
    AnalogWaveformCollection<double> measurementWaveforms
)
```

```
Public Function FetchArrayMeasurement (
    timeout As PrecisionTimeSpan,
    arrayMeasurementType As ScopeArrayMeasurementType,
    measurementWaveforms As AnalogWaveformCollection(Of Double)
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The type of the array measurement to perform.

measurementWaveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `measurementWaveforms` to `.` The method will also allocate memory during the call if `measurementWaveforms` is set to a waveform object with zero-sized data or `measurementWaveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if

measurementWaveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from measurementWaveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

Exceptions

Exception	Condition
ObjectDisposedException	FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value of arrayMeasurementType was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchArrayMeasurement Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchArrayMeasurement Method
(PrecisionTimeSpan, ScopeArrayMeasurementType,
AnalogWaveformCollectionDouble)

Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchArrayMeasurement(
    PrecisionTimeSpan timeout,
    ScopeArrayMeasurementType arrayMeasurementType,
    AnalogWaveformCollection<double> measurementWaveforms
)
```

```
Public Function FetchArrayMeasurement (
    timeout As PrecisionTimeSpan,
    arrayMeasurementType As ScopeArrayMeasurementType,
    measurementWaveforms As AnalogWaveformCollection(Of Double)
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The type of the array measurement to perform.

measurementWaveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `measurementWaveforms` to `.` The method will also allocate memory during the call if `measurementWaveforms` is set to a waveform object with zero-sized data or `measurementWaveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `measurementWaveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `measurementWaveforms` will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An array of type [AnalogWaveformCollectionTData](#).

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value of <code>arrayMeasurementType</code> was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

FetchArrayMeasurement Overload

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchArrayMeasurement Method
(PrecisionTimeSpan, ScopeArrayMeasurementType,
AnalogWaveformCollectionDouble, ScopeWaveformInfo)

Obtains a waveform from the digitizer and returns the specified measurement array. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchArrayMeasurement(  
    PrecisionTimeSpan timeout,  
    ScopeArrayMeasurementType arrayMeasurementType,  
    AnalogWaveformCollection<double> measurementWaveforms,  
    out ScopeWaveformInfo[] waveformInfo  
)
```

```
Public Function FetchArrayMeasurement (  
    timeout As PrecisionTimeSpan,  
    arrayMeasurementType As ScopeArrayMeasurementType,  
    measurementWaveforms As AnalogWaveformCollection(Of Double),  
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()  
) As AnalogWaveformCollection(Of Double)
```


Parameters

timeout

Type: `NationalInstruments.PrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

arrayMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeArrayMeasurementType](#)

The type of the array measurement to perform.

measurementWaveforms

Type: `NationalInstruments.AnalogWaveformCollectionDouble`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `measurementWaveforms` to `.` The method also allocates memory during the call if `measurementWaveforms` is set to a waveform object with zero-sized data or `waveformInfo` is set to a waveform object with a different size than that required for the output waveform. For example, `measurementWaveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `measurementWaveforms` will be reused for the output waveform.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects containing the timing and scaling information.

Return Value

Type: `AnalogWaveformCollectionDouble`

An array of type `AnalogWaveformCollectionTData`.

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchArrayMeasurement(PrecisionTimeSpan, ScopeArrayMeasurementType, AnalogWaveformCollectionDouble,</code>

	ScopeWaveformInfo) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value of arrayMeasurementType was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)



[FetchArrayMeasurement Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchByte Method

Overload List

Name	Description
 FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)	Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte, ScopeWaveformInfo)	Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify. Refer to Features Supported by Device for more information on using this method.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchByte Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)

Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<byte> FetchByte(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<byte> waveforms  
)
```

```
Public Function FetchByte (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Byte)  
) As AnalogWaveformCollection(Of Byte)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionByte`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `new AnalogWaveformCollectionByte()`. The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

Return Value

Type: `AnalogWaveformCollectionByte`

An object of type `AnalogWaveformCollectionTData`.

`AnalogWaveformCollectionTData` returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index $2x$ = record 1, channel 0

index $3x$ = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Using Fetch Functions](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchByte Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte)

Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<byte> FetchByte(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<byte> waveforms  
)
```

```
Public Function FetchByte (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Byte)  
) As AnalogWaveformCollection(Of Byte)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using MaxValue for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0

was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollection`[Byte](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

Return Value

Type: `AnalogWaveformCollection`[Byte](#)

An object of type `AnalogWaveformCollectionTData`.

`AnalogWaveformCollectionTData` returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `.`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to , then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Using Fetch Functions](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchByte Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte, ScopeWaveformInfo)

Retrieves data from a previously initiated acquisition and returns binary 8-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify. Refer to [Features Supported by Device](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<byte> FetchByte(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<byte> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchByte (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Byte),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Byte)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever data is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollection`[Byte](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set

to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionByte](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Return Value

Type: [AnalogWaveformCollectionByte](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchByte(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte, ScopeWaveformInfo) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Using Fetch Functions](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)



[FetchByte Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexDouble Method

Overload List

Name	Description
 FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)	Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.
 FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble, ScopeWaveformInfo)	Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexDouble Me)

Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexDouble> FetchComplexDouble (
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexDouble> waveforms
)
```

```
Public Function FetchComplexDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexDouble)
) As ComplexWaveformCollection(Of ComplexDouble)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexDouble`

An array of type `ComplexWaveformCollectionTData`.

Return Value

Type: `ComplexWaveformCollectionComplexDouble`

An array of type `ComplexWaveformCollectionTData`

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
<code>IviCDriverException</code>	<p>The channel specified is not enabled.</p> <p>- or -</p> <p>The session is not configured to fetch complex measurements.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchComplexDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexDouble Method
(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)

Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexDouble> FetchComplexDouble(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    ComplexWaveformCollection<ComplexDouble> waveforms  
)
```

```
Public Function FetchComplexDouble (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As ComplexWaveformCollection(Of ComplexDouble)  
) As ComplexWaveformCollection(Of ComplexDouble)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexDouble`

An array of type `ComplexWaveformCollectionTData`.

Return Value

Type: `ComplexWaveformCollectionComplexDouble`

An array of type `ComplexWaveformCollectionTData`

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
<code>IviCDriverException</code>	The channel specified is not enabled. - or - The session is not configured to fetch complex measurements. - or - The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchComplexDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexDouble Method
(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble,
ScopeWaveformInfo)

Retrieves data that the digitizer has acquired from a previously initiated acquisition and returns a one-dimensional array of complex, scaled waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexDouble> FetchComplexDouble (
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexDouble> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchComplexDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexDouble),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As ComplexWaveformCollection(Of ComplexDouble)
```


Parameters

timeout

Type: `NationalInstruments.PrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [System.Int64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstruments.ComplexWaveformCollectionComplexDouble`

An array whose length is the numSamples times number of waveforms.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of structures with the following timing and scaling information

Return Value

Type: `ComplexWaveformCollectionComplexDouble`

An array of type `ComplexWaveformCollectionTData`.

Return Value

Type: `ComplexWaveformCollectionComplexDouble`

An array of [ScopeWaveformInfo](#) objects with the timing and scaling information.

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchComplexDouble(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexDouble, ScopeWaveformInfo)</code>

	was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The channel specified is not enabled. - or - The session is not configured to fetch complex measurements. - or - The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)



[FetchComplexDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexInt16 Method

Overload List

Name	Description
 FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16)	Retrieves data from single channels and records. A one-dimensional array of complex binary 16-bit waveforms.
 FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16, ScopeWaveformInfo)	Retrieves data from single channels and records. Returns a one-dimensional array of complex binary 16-bit waveforms.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexInt16 Method

Retrieves data from single channels and records. A one-dimensional array of complex binary 16-bit waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexInt16> FetchComplexInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexInt16> waveforms
)
```

```
Public Function FetchComplexInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexInt16)
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the

acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

Return Value

Type: `ComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

Exceptions

Exception	Condition
ObjectDisposedException	FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The channel specified is not enabled. - or - The session is not configured to fetch complex measurements. - or - The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexInt16 Method (PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16)

Retrieves data from single channels and records. A one-dimensional array of complex binary 16-bit waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexInt16> FetchComplexInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexInt16> waveforms
)
```

```
Public Function FetchComplexInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexInt16)
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples.

The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

Return Value

Type: `ComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

Exceptions

Exception	Condition
ObjectDisposedException	FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The channel specified is not enabled. - or - The session is not configured to fetch complex measurements. - or - The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchComplexInt16 Method
(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16, ScopeWaveformInfo)

Retrieves data from single channels and records. Returns a one-dimensional array of complex binary 16-bit waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ComplexWaveformCollection<ComplexInt16> FetchComplexInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    ComplexWaveformCollection<ComplexInt16> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchComplexInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As ComplexWaveformCollection(Of ComplexInt16),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As ComplexWaveformCollection(Of ComplexInt16)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all

available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
An array of [ScopeWaveformInfo](#) objects with the timing and scaling information

Return Value

Type: `ComplexWaveformCollectionComplexInt16`
An array of type `ComplexWaveformCollectionTData`.

Exceptions

Exception	Condition
ObjectDisposedException	FetchComplexInt16(PrecisionTimeSpan, Int64, ComplexWaveformCollectionComplexInt16, ScopeWaveformInfo) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The channel specified is not enabled. - or - The session is not configured to fetch complex measurements. - or - The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)



[FetchComplexInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchDouble Method

Overload List

Name	Description
 FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)	Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to Using Fetch Functions for more information on using this method.
 FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)	Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to Using Fetch Functions for more information on using this method.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchDouble Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)

Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to [Using Fetch Functions](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchDouble(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<double> waveforms
)
```

```
Public Function FetchDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Double)
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output

waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to , then the waveforms returned will contain twice the number of samples specified by numberOfSamples. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to , then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchDouble (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble) was called after the associated NIScope or ScopeDriverUtility object was disposed.

IviDriverException

The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchDouble Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)

Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to [Using Fetch Functions](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchDouble(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<double> waveforms
)
```

```
Public Function FetchDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
```

```

waveforms As AnalogWaveformCollection(Of Double)
) As AnalogWaveformCollection(Of Double)

```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0
 index x = record 0, channel 1
 index 2x = record 1, channel 0
 index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to , then the waveforms returned will contain twice the number of samples specified by numberOfSamples. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to , then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchDouble (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchDouble Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchDouble Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)

Returns the waveform from a previously initiated acquisition that the digitizer acquires for the specified channel. This method returns scaled voltage waveforms. Refer to [Using Fetch Functions](#) for more information on using this method.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> FetchDouble(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<double> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchDouble (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Double),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the

acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contain timing and scaling information about each waveform

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length



Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchDouble(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
<code>IviDriverException</code>	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchDouble Overload](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt16 Method

Overload List

Name	Description
------	-------------

 FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16)	Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16, ScopeWaveformInfo)	Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt16 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16)

Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public AnalogWaveformCollection<short> FetchInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<short> waveforms
)
```

```
Public Function FetchInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Short)
) As AnalogWaveformCollection(Of Short)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionInt16`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output

waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionInt16](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `0`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `1`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt16 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16)

Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<short> FetchInt16(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<short> waveforms  
)
```

```
Public Function FetchInt16 (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Short)  
) As AnalogWaveformCollection(Of Short)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionInt16`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `new AnalogWaveformCollectionInt16()`. The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

Return Value

Type: `AnalogWaveformCollectionInt16`

An object of type `AnalogWaveformCollectionTData`. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index $2x$ = record 1, channel 0
 index $3x$ = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInt16 (PrecisionTimeSpan , Int64 , AnalogWaveformCollectionInt16) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt16 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt16 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16, ScopeWaveformInfo)

Retrieves data from a previously initiated acquisition and returns binary 16-bit waveforms. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<short> FetchInt16(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<short> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchInt16 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Short),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Short)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using MaxValue for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the

acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionInt16](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionInt16](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length



Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>FetchInt16(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16, ScopeWaveformInfo)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt16 Overload](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt32 Method

Overload List

Name	Description
------	-------------

 FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32)	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32, ScopeWaveformInfo)	Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)[See Also](#)[Reference](#)[ScopeChannelMeasurement Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt32 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32)

Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

[Syntax](#)[C#VB](#)

```
public AnalogWaveformCollection<int> FetchInt32 (
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<int> waveforms
)
```

```
Public Function FetchInt32 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Integer)
) As AnalogWaveformCollection(Of Integer)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use `-1` for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionInt32](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output

waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `0`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `1`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInt32(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt32 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32)

Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<int> FetchInt32(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<int> waveforms  
)
```

```
Public Function FetchInt32 (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Integer)  
) As AnalogWaveformCollection(Of Integer)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using `MaxValue` for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionInt32`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `new AnalogWaveformCollectionInt32()`. The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

Return Value

Type: `AnalogWaveformCollectionInt32`

An object of type `AnalogWaveformCollectionTData`. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index $2x$ = record 1, channel 0

index $3x$ = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInt32 (PrecisionTimeSpan , Int64 , AnalogWaveformCollectionInt32) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchInt32 Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32, ScopeWaveformInfo)

Retrieves data from a previously initiated acquisition. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<int> FetchInt32(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<int> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function FetchInt32 (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Integer),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Integer)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using MaxValue for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the

acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method throws an exception.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionInt32](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)

An array of [ScopeWaveformInfo](#) objects that contains timing and scaling information about each waveform.

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Return Value

Type: [AnalogWaveformCollectionInt32](#)

An array of [ScopeWaveformInfo](#) objects that contain timing and scaling information about each waveform

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInt32 (PrecisionTimeSpan , Int64 , AnalogWaveformCollectionInt32 , ScopeWaveformInfo) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Some functionality, such as time stamping, is not supported in all digitizers. Refer to [Features Supported by Device](#) for more information.

Refer to [Using Fetch Functions](#) for more information on using this method.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[FetchInt32 Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchScalarMeasurement Method

Fetches a waveform from the digitizer and performs the specified waveform measurement. Refer to [Using Fetch Functions](#) for more information.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double[] FetchScalarMeasurement (
    PrecisionTimeSpan timeout,
    ScopeScalarMeasurementType scalarMeasurementType
)
```

```
Public Function FetchScalarMeasurement (
    timeout As PrecisionTimeSpan,
    scalarMeasurementType As ScopeScalarMeasurementType
) As Double()
```

Parameters

timeout

Type: [NationalInstruments.PrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScope.ScopeScalarMeasurementType](#)

The type of the scalar measurement to be performed.

Return Value

Type: [Double](#)

An array of all measurements acquired.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFetchScalarMeasurementStatistics Method

Obtains a waveform measurement and returns the measurement value. This method may return multiple statistical results depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeScalarMeasurementStatistics[] FetchScalarMeasurementStatistics (
    PrecisionTimeSpan timeout,
    ScopeScalarMeasurementType scalarMeasurementType
)
```

```
Public Function FetchScalarMeasurementStatistics (
    timeout As PrecisionTimeSpan,
    scalarMeasurementType As ScopeScalarMeasurementType
) As ScopeScalarMeasurementStatistics()
```

Parameters

timeout

Type: NationalInstrumentsPrecisionTimeSpan

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The type of the scalar measurement to be performed on each fetched waveform.

Return Value

Type: [ScopeScalarMeasurementStatistics](#)

An array of type [ScopeScalarMeasurementStatistics](#).

See Also

Reference



[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementRead Method

Overload List

Name	Description
 Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.
 Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)	Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

[Top](#)

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementRead Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)

Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> Read(  
    PrecisionTimeSpan timeout,  
    long numberOfSamples,  
    AnalogWaveformCollection<double> waveforms  
)
```

```
Public Function Read (  
    timeout As PrecisionTimeSpan,  
    numberOfSamples As Long,  
    waveforms As AnalogWaveformCollection(Of Double)  
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: NationalInstrumentsPrecisionTimeSpan

The time to wait for data to be acquired. Using 0 for this parameter tells NI-

SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionTData](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

 Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `numberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[Read Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

`ScopeChannelMeasurementRead` Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)

Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> Read(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<double> waveforms
)
```

```
Public Function Read (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Double)
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: [SystemInt64](#)

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: [NationalInstrumentsAnalogWaveformCollectionDouble](#)

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set waveforms to . The method will also allocate memory during the call if waveforms is set to a waveform object with zero-sized data or waveforms is set to a waveform

object with a different size than that required for the output waveform. For example, if waveforms is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from waveforms will be reused for the output waveform.

Return Value

Type: [AnalogWaveformCollectionDouble](#)

An object of type [AnalogWaveformCollectionTData](#). NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0

index x = record 0, channel 1

index 2x = record 1, channel 0

index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[Read Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementRead Method (PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)

Initiates an acquisition, waits for it to complete, and retrieves the data. This method may return multiple waveforms depending on the number of channels, the acquisition type, and the number of records you specify.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public AnalogWaveformCollection<double> Read(
    PrecisionTimeSpan timeout,
    long numberOfSamples,
    AnalogWaveformCollection<double> waveforms,
    out ScopeWaveformInfo[] waveformInfo
)
```

```
Public Function Read (
    timeout As PrecisionTimeSpan,
    numberOfSamples As Long,
    waveforms As AnalogWaveformCollection(Of Double),
    <OutAttribute> ByRef waveformInfo As ScopeWaveformInfo()
) As AnalogWaveformCollection(Of Double)
```

Parameters

timeout

Type: `NationalInstrumentsPrecisionTimeSpan`

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

numberOfSamples

Type: `SystemInt64`

The maximum number of samples to fetch for each waveform. If the acquisition finishes with fewer points than requested, some instruments return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use -1 for this parameter if you want to fetch all available samples. The method reads the actual record length and attempts to acquire all available samples. If it fails to complete within the timeout period, the method returns an error.

waveforms

Type: `NationalInstrumentsAnalogWaveformCollectionDouble`

The waveform object whose memory can be reused while creating the output waveform. To allocate memory during the call to this method, set `waveforms` to `.` The method will also allocate memory during the call if `waveforms` is set to a waveform object with zero-sized data or `waveforms` is set to a waveform object with a different size than that required for the output waveform. For example, if `waveforms` is set to a waveform object with the same size as that required for this method call, then no new memory is allocated for the output waveform, and instead the memory of the waveform object from `waveforms` will be reused for the output waveform.

waveformInfo

Type: `NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo`

An array of `ScopeWaveformInfo` objects.

Return Value

Type: `AnalogWaveformCollectionDouble`

An object of type `AnalogWaveformCollectionTData`. NI-SCOPE returns this data sequentially, so all record 0 waveforms are first. For example, with a channel list of 0,1, you would have the following index values:

index 0 = record 0, channel 0
 index x = record 0, channel 1
 index 2x = record 1, channel 0
 index 3x = record 1, channel 1

Where x = the record length

Note

If you are using this entry point or any of its overloads or async versions to read complex data, then you will either receive double the number of samples or double the number of records requested depending on the value of [FetchInterleavedIQData](#).

If [FetchInterleavedIQData](#) is set to `True`, then the waveforms returned will contain twice the number of samples specified by `NumberOfSamples`. Every two samples represents interleaved IQ data.

If [FetchInterleavedIQData](#) is set to `False`, then the waveform collection returned will contain twice the number of records requested. Each pair of records represents one waveform of I data, and one waveform of Q data.

Exceptions

Exception	Condition
ObjectDisposedException	<code>Read(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble, ScopeWaveformInfo)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[Read Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReadScalarMeasurement Method

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record or for multiple channels and records. Refer to [Using Fetch Functions](#) for more information.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double[] ReadScalarMeasurement(  
    PrecisionTimeSpan timeout,  
    ScopeScalarMeasurementType scalarMeasurementType  
)
```

```
Public Function ReadScalarMeasurement (  
    timeout As PrecisionTimeSpan,  
    scalarMeasurementType As ScopeScalarMeasurementType  
) As Double()
```

Parameters

timeout

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The time to wait for data to be acquired. Using 0 for this parameter tells NI-SCOPE to fetch whatever is currently available. Using -1 for this parameter implies infinite timeout.

scalarMeasurementType

Type: [NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementType](#)

The type of the scalar measurement to be performed.

Return Value

Type: [Double](#)

An array of all measurements acquired.

See Also

Reference

[ScopeChannelMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Class

Represents the [ScopeChannelMeasurement](#) sub-objects that contain properties related to filtering.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementFilter](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB










```
public sealed class ScopeChannelMeasurementFilter : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelMeasurementFilter
    Inherits ScopeSubObject
```

The `ScopeChannelMeasurementFilter` type exposes the following members.





Properties

Name	Description
------	-------------

 CenterFrequency	Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by Width , where the cutoff frequencies are the center width.
 CutoffFrequency	Gets or sets the cutoff frequency, in hertz, for lowpass and highpass filters. The cutoff frequency definition varies depending on the filter.
 FirFilterWindow	Gets or sets the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.
 Order	Gets or sets the order of an IIR filter. All positive integers are valid.
 Ripple	Gets or sets the amount of passband ripple, in decibels (positive values), for Chebyshev filters.
 Taps	Gets or sets the number of taps (coefficients) for a FIR filter.
 TransientPercent	Gets or sets the percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This process eliminates the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.
 Type	Gets or sets the type of filter, for both IIR and FIR filters.
 Width	Gets or sets the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at CenterFrequency \pm one-half width.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Digital Filtering and Filter Measurements topics in the [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources








[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Properties

The [ScopeChannelMeasurementFilter](#) type exposes the following members.

Properties

Name	Description
 CenterFrequency	Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by Width , where the cutoff frequencies are the center width.
 CutoffFrequency	Gets or sets the cutoff frequency, in hertz, for lowpass and highpass filters. The cutoff frequency definition varies depending on the filter.
 FirFilterWindow	Gets or sets the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.
 Order	Gets or sets the order of an IIR filter. All positive integers are valid.
 Ripple	Gets or sets the amount of passband ripple, in decibels (positive values), for Chebyshev filters.
 Taps	Gets or sets the number of taps (coefficients) for a FIR filter.
 TransientPercent	Gets or sets the percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This process eliminates the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.

 Type	Gets or sets the type of filter, for both IIR and FIR filters.
 Width	Gets or sets the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at CenterFrequency ±one-half width.

[Top](#)

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCenterFrequency Property

Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by [Width](#), where the cutoff frequencies are the center width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCenterFrequency Property

Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by [Width](#), where the cutoff frequencies are the center width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCutoffFrequency Property

Gets or sets the cutoff frequency, in hertz, for lowpass and highpass filters. The cutoff frequency definition varies depending on the filter.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CutoffFrequency { get; set; }
```

```
Public Property CutoffFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterFirFilterWindow Property

Gets or sets the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementFirFilterWindow FirFilterWindow { get; set; }
```

```
Public Property FirFilterWindow As ScopeMeasurementFirFilterWindow  
    Get  
    Set
```

Property Value

Type: [ScopeMeasurementFirFilterWindow](#)

Specifies the [ScopeMeasurementFirFilterWindow](#). The default value is [None](#).

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterOrder Property

Gets or sets the order of an IIR filter. All positive integers are valid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Order { get; set; }
```

```
Public Property Order As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 2.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterRipple Property

Gets or sets the amount of passband ripple, in decibels (positive values), for Chebyshev filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Ripple { get; set; }
```

```
Public Property Ripple As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0.1 dB.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterTaps Property

Gets or sets the number of taps (coefficients) for a FIR filter.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Taps { get; set; }
```

```
Public Property Taps As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 25.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterTransientPercent P

Gets or sets the percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This process eliminates the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double TransientPercent { get; set; }
```

```
Public Property TransientPercent As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 20.0%.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterType Property

Gets or sets the type of filter, for both IIR and FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementFilterType Type { get; set; }
```

```
Public Property Type As ScopeMeasurementFilterType  
    Get  
    Set
```

Property Value

Type: [ScopeMeasurementFilterType](#)

Specifies the [ScopeMeasurementFilterType](#). The default value is [LowPass](#).

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterWidth Property

Gets or sets the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at [CenterFrequency](#) \pm one-half width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Width { get; set; }
```

```
Public Property Width As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e3 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources










[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Properties

The [ScopeChannelMeasurementFilter](#) type exposes the following members.

Properties

Name	Description
 CenterFrequency	Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by Width , where the cutoff frequencies are the center width.
 CutoffFrequency	Gets or sets the cutoff frequency, in hertz, for lowpass and highpass filters. The cutoff frequency definition varies depending on the filter.
 FirFilterWindow	Gets or sets the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.
 Order	Gets or sets the order of an IIR filter. All positive integers are valid.
 Ripple	Gets or sets the amount of passband ripple, in decibels (positive values), for Chebyshev filters.
 Taps	Gets or sets the number of taps (coefficients) for a FIR filter.
 TransientPercent	Gets or sets the percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This process eliminates the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.
 Type	Gets or sets the type of filter, for both IIR and FIR filters.
 Width	Gets or sets the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at CenterFrequency ±one-half width.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCenterFrequency Property

Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by [Width](#), where the cutoff frequencies are the center width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCenterFrequency Property

Gets or sets the center frequency, in hertz, for bandpass and bandstop filters. The width of the filter is specified by [Width](#), where the cutoff frequencies are the center width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CenterFrequency { get; set; }
```

```
Public Property CenterFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterCutoffFrequency Property

Gets or sets the cutoff frequency, in hertz, for lowpass and highpass filters. The cutoff frequency definition varies depending on the filter.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double CutoffFrequency { get; set; }
```

```
Public Property CutoffFrequency As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e6 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterFirFilterWindow Property

Gets or sets the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementFirFilterWindow FirFilterWindow { get; set; }
```

```
Public Property FirFilterWindow As ScopeMeasurementFirFilterWindow  
    Get  
    Set
```

Property Value

Type: [ScopeMeasurementFirFilterWindow](#)

Specifies the [ScopeMeasurementFirFilterWindow](#). The default value is [None](#).

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterOrder Property

Gets or sets the order of an IIR filter. All positive integers are valid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Order { get; set; }
```

```
Public Property Order As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 2.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterRipple Property

Gets or sets the amount of passband ripple, in decibels (positive values), for Chebyshev filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Ripple { get; set; }
```

```
Public Property Ripple As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 0.1 dB.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterTaps Property

Gets or sets the number of taps (coefficients) for a FIR filter.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Taps { get; set; }
```

```
Public Property Taps As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is 25.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterTransientPercent Property

Gets or sets the percentage (0 - 100%) of the IIR filtered waveform to eliminate from the beginning of the waveform. This process eliminates the transient portion of the waveform that is undefined due to the assumptions necessary at the boundary condition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double TransientPercent { get; set; }
```

```
Public Property TransientPercent As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 20.0%.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterType Property

Gets or sets the type of filter, for both IIR and FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementFilterType Type { get; set; }
```

```
Public Property Type As ScopeMeasurementFilterType
    Get
    Set
```

Property Value

Type: [ScopeMeasurementFilterType](#)

Specifies the [ScopeMeasurementFilterType](#). The default value is [LowPass](#).

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilterWidth Property

Gets or sets the width of bandpass and bandstop type filters in hertz. The cutoff frequencies occur at [CenterFrequency](#) ±one-half width.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Width { get; set; }
```

```
Public Property Width As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 1.0e3 Hz.

See Also

Reference

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementFilter Methods

The [ScopeChannelMeasurementFilter](#) type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

[See Also](#)

[Reference](#)

[ScopeChannelMeasurementFilter Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Class

Represents the [ScopeChannelMeasurement](#) sub-objects that contain properties related to reference levels.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementReferenceLevel](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax






[C#](#)[VB](#)

```
public sealed class ScopeChannelMeasurementReferenceLevel : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelMeasurementReferenceLevel  
    Inherits ScopeSubObject
```





The `ScopeChannelMeasurementReferenceLevel` type exposes the following members.

Properties

Name	Description
 High	Gets or sets the high reference level used in many scalar measurements.
 Low	Gets or sets the low reference level used in many scalar measurements.
 Mid	Gets or sets the mid reference level used in many scalar measurements.
 PercentageMethod	Gets or sets the mode used to map percentage reference units to voltages for the reference, when Units is set to Percentage .
 Units	Gets or sets the units for the waveform measurement reference levels. You can use this property to specify whether you are providing the actual threshold voltage values or percentage values.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Reference Levels topic in the [NI High-Speed Digitizers Help](#) or the Attributes section in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources






[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Properties

The [ScopeChannelMeasurementReferenceLevel](#) type exposes the following members.

Properties

Name	Description
 High	Gets or sets the high reference level used in many scalar measurements.
 Low	Gets or sets the low reference level used in many scalar measurements.
 Mid	Gets or sets the mid reference level used in many scalar measurements.
 PercentageMethod	Gets or sets the mode used to map percentage reference units to voltages for the reference, when Units is set to Percentage .
 Units	Gets or sets the units for the waveform measurement reference levels. You can use this property to specify whether you are providing the actual threshold voltage values or percentage values.

[Top](#)

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelHigh Property

Gets or sets the high reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The high reference level used in many scalar measurements. The default value is 90%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelHigh Prop

Gets or sets the high reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The high reference level used in many scalar measurements. The default value is 90%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelLow Prop

Gets or sets the low reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Low { get; set; }
```

```
Public Property Low As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The low reference level used in many scalar measurements. The default value is 10%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelMid Property

Gets or sets the mid reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Mid { get; set; }
```

```
Public Property Mid As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The mid reference level used in many scalar measurements. The default value is 50%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelPercentage

Gets or sets the mode used to map percentage reference units to voltages for the reference, when [Units](#) is set to [Percentage](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementPercentageMethod PercentageMethod { get; set; }
```

```
Public Property PercentageMethod As ScopeMeasurementPercentageMethod  
    Get  
    Set
```

Property Value

Type: [ScopeMeasurementPercentageMethod](#)

Specifies the [ScopeMeasurementPercentageMethod](#). The default value of this property is [BaseTop](#).

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelUnits Pro

Gets or sets the units for the waveform measurement reference levels. You can use this property to specify whether you are providing the actual threshold voltage values or percentage values.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementReferenceLevelUnits Units { get; set; }
```

```
Public Property Units As ScopeMeasurementReferenceLevelUnits
    Get
    Set
```

Property Value

Type: [ScopeMeasurementReferenceLevelUnits](#)

Specifies the [ScopeMeasurementReferenceLevelUnits](#). The default value is [Percentage](#).

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Properties

The [ScopeChannelMeasurementReferenceLevel](#) type exposes the following members.

Properties

Name	Description
 High	Gets or sets the high reference level used in many scalar measurements.
 Low	Gets or sets the low reference level used in many scalar measurements.
 Mid	Gets or sets the mid reference level used in many scalar measurements.
 PercentageMethod	Gets or sets the mode used to map percentage reference units to voltages for the reference, when Units is set to Percentage .

 **Units**

Gets or sets the units for the waveform measurement reference levels. You can use this property to specify whether you are providing the actual threshold voltage values or percentage values.

[Top](#)

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelHigh Prop

Gets or sets the high reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The high reference level used in many scalar measurements. The default value is 90%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelHigh Property

Gets or sets the high reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The high reference level used in many scalar measurements. The default value is 90%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelLow Property

Gets or sets the low reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Low { get; set; }
```

```
Public Property Low As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The low reference level used in many scalar measurements. The default value is 10%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelMid Property

Gets or sets the mid reference level used in many scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Mid { get; set; }
```

```
Public Property Mid As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The mid reference level used in many scalar measurements. The default value is 50%.

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelPercentageMethod Property

Gets or sets the mode used to map percentage reference units to voltages for the reference, when [Units](#) is set to [Percentage](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementPercentageMethod PercentageMethod { get; set; }
```

```
Public Property PercentageMethod As ScopeMeasurementPercentageMethod
    Get
    Set
```

Property Value

Type: [ScopeMeasurementPercentageMethod](#)

Specifies the [ScopeMeasurementPercentageMethod](#). The default value of this property is [BaseTop](#).

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevelUnits Property

Gets or sets the units for the waveform measurement reference levels. You can use this property to specify whether you are providing the actual threshold voltage values or percentage values.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeMeasurementReferenceLevelUnits Units { get; set; }
```

```
Public Property Units As ScopeMeasurementReferenceLevelUnits
    Get
    Set
```

Property Value

Type: [ScopeMeasurementReferenceLevelUnits](#)

Specifies the [ScopeMeasurementReferenceLevelUnits](#). The default value is [Percentage](#).

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementReferenceLevel Methods

The [ScopeChannelMeasurementReferenceLevel](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)

 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelMeasurementReferenceLevel Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Class

Provides the properties related to time histograms, which place acquired samples into bins based on their time relative to a trigger point.

Inheritance Hierarchy

[SystemObject](#)[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementTimeHistogram

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax






[C#VB](#)

```
public sealed class ScopeChannelMeasurementTimeHistogram : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelMeasurementTimeHistogram
    Inherits ScopeSubObject
```





The `ScopeChannelMeasurementTimeHistogram` type exposes the following members.

Properties

Name	Description
 HighTime	Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition time histogram.
 LowTime	Gets or sets the lowest time value, in seconds, included in the multiple-acquisition time histogram.
 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition time histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the [Time Histograms Overview](#) and the [Time Histogram Measurements](#) topics in the [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Properties

The [ScopeChannelMeasurementTimeHistogram](#) type exposes the following members.

Properties

Name	Description
 HighTime	Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition time histogram.
 LowTime	Gets or sets the lowest time value, in seconds, included in the multiple-acquisition time histogram.
 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition time histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

[See Also](#)

[Reference](#)

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighTime

Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan HighTime { get; set; }
```

```
Public Property HighTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is 5.0e-4 seconds.

[See Also](#)

[Reference](#)

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighTime

Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan HighTime { get; set; }
```

```
Public Property HighTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is 5.0e-4 seconds.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighVolts

Gets or sets the highest voltage value included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramLowTime

Gets or sets the lowest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan LowTime { get; set; }
```

```
Public Property LowTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is $-5.0e-4$ seconds.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramLowVolts

Gets or sets the lowest voltage value included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowVolts { get; set; }
```

```
Public Property LowVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is -10.0 V.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramSize Property

Gets or sets the multiple-acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with [ClearWaveformMeasurements](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Size { get; set; }
```

```
Public Property Size As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Properties

The [ScopeChannelMeasurementTimeHistogram](#) type exposes the following members.

Properties

Name	Description
 HighTime	Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition time histogram.
 LowTime	Gets or sets the lowest time value, in seconds, included in the multiple-acquisition time histogram.

 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition time histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighTime

Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public PrecisionTimeSpan HighTime { get; set; }
```

```
Public Property HighTime As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is 5.0e-4 seconds.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighTime Property

Gets or sets the highest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan HighTime { get; set; }
```

```
Public Property HighTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is 5.0e-4 seconds.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramHighVolts Property

Gets or sets the highest voltage value included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramLowTime Property

Gets or sets the lowest time value, in seconds, included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan LowTime { get; set; }
```

```
Public Property LowTime As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The default value is $-5.0e-4$ seconds.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramLowVolts Property

Gets or sets the lowest voltage value included in the multiple-acquisition time histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowVolts { get; set; }
```

```
Public Property LowVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is -10.0 V.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogramSize Property

Gets or sets the multiple-acquisition voltage histogram size. The size is set during the first call to a time histogram measurement after you clear the measurement history with [ClearWaveformMeasurements](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Size { get; set; }
```

```
Public Property Size As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementTimeHistogram Methods

The [ScopeChannelMeasurementTimeHistogram](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object.

(Inherited from [Object](#).)

[Top](#)

See Also

Reference

[ScopeChannelMeasurementTimeHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Class

Represents the [ScopeChannelMeasurement](#) sub-objects that contain properties related to voltage histograms.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementVoltageHistogram](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB

```
public sealed class ScopeChannelMeasurementVoltageHistogram : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelMeasurementVoltageHistogram
    Inherits ScopeSubObject
```





The `ScopeChannelMeasurementVoltageHistogram` type exposes the following members.

Properties

Name	Description
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.
 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition voltage histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Voltage Histograms Overview and the Voltage Histogram Measurements topics in the [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Properties

The [ScopeChannelMeasurementVoltageHistogram](#) type exposes the following members.

Properties

Name	Description
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.
 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition voltage histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramHighVolts

Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramHighVolts

Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramLowVolts

Gets or sets the lowest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowVolts { get; set; }
```

```
Public Property LowVolts As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is -10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramSize Property

Gets or sets the multiple-acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after you clear the measurement history with [ClearWaveformMeasurements](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Size { get; set; }
```

```
Public Property Size As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Properties

The [ScopeChannelMeasurementVoltageHistogram](#) type exposes the following members.

Properties

Name	Description
 HighVolts	Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.
 LowVolts	Gets or sets the lowest voltage value included in the multiple-acquisition voltage histogram.
 Size	Gets or sets the multiple-acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after you clear the measurement history with ClearWaveformMeasurements .

[Top](#)

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramHighVolts

Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramHighVolts Property

Gets or sets the highest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighVolts { get; set; }
```

```
Public Property HighVolts As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The default value is 10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramLowVolts Property

Gets or sets the lowest voltage value included in the multiple-acquisition voltage histogram.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowVolts { get; set; }
```

```
Public Property LowVolts As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The default value is -10.0 V.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogramSize Property

Gets or sets the multiple-acquisition voltage histogram size. The size is set the first time a voltage histogram measurement is called after you clear the measurement history with [ClearWaveformMeasurements](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long Size { get; set; }
```

```
Public Property Size As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is 256.

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelMeasurementVoltageHistogram Methods

The [ScopeChannelMeasurementVoltageHistogram](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelMeasurementVoltageHistogram Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Class

Provides methods and properties for configuring the Onboard Signal Processing (OSP) components of high speed digitizers.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeChannelOnboardSignalProcessing

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public sealed class ScopeChannelOnboardSignalProcessing : ScopeSubObject
```

```
Public NotInheritable Class ScopeChannelOnboardSignalProcessing
    Inherits ScopeSubObject
```

The ScopeChannelOnboardSignalProcessing type exposes the following members.





Properties

Name	Description
 Ddc	Gets the ScopeChannelDdc sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.
 EqualizationFilter	Gets the ScopeChannelEqualizationFilter sub-object that is related to configuring the Equalization Filter onboard the high-speed digitizer.

Top

Methods

Name	Description
------	-------------

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

This property can be used only with high-speed digitizers that support onboard signal processing (OSP). `ModularInstrumentsException` is thrown when this property is accessed or any of the sub-properties are set or accessed on an instrument that does not provide OSP. The main components of the Onboard Signal Processing block include the following:

- Digital Down-Converter - [ScopeChannelDdc](#)
The DDC takes a programmable portion of the Nyquist_Theorem spectrum of the ADC and moves it down to baseband (centered around 0 Hz). This process is performed by first frequency translating the center of the band of interest to 0 Hz. The data is then filtered and decimated to get alias-protected data of the requested bandwidth. The data output from the DDC is stored into memory at this lower sample rate as complex IQ pairs.
- Filtering And Decimation - [ScopeChannelEqualizationFilter](#)
The filtering and decimation stage of the OSP block reduces the effective sample rate of the digitizer while protecting the frequency spectrum of the decimated data from aliases. This protection occurs when the data passes through a lowpass filter before decimation. The normal decimation in digitizers (when not using OSP) does not protect the frequency spectrum of the decimated data from aliases.
- Digital Gain - [DigitalGain](#)
Digital gain can be used to change the gain of each channel. You can program the digital gain of each channel independently using the [DigitalGain](#). The gain

can range from -1.5 to $+1.5$ (unitless). You can use the digital gain to attenuate the channel data to eliminate overflows in the OSP block.

- Digital Offset - [DigitalOffset](#)

The digital offset can be used to change the offset of each channel. You can program the digital offset of each channel independently using the [DigitalOffset](#).

- Frequency Translation - [FrequencyTranslationEnabled](#)

Frequency translation moves the Nyquist Theorem frequency spectrum of the acquired signal by the amount specified by the [CenterFrequency](#). This is accomplished by multiplying the acquired data stream with the output of the numerically controlled oscillator (NCO). The NCO is a digital circuit that creates two sine waves of the same frequency (the Center Frequency) with two independent phases. You can use the CenterFrequency to set the carrier frequency. The I phase cosine waveform is multiplied by the I signal path, and the Q phase cosine waveform is multiplied by the Q data path. The I and Q phases are programmable from 0° to 360° by setting the [FrequencyTranslationPhaseI](#) and [FrequencyTranslationPhaseQ](#) properties. Both the frequency and the phases can be updated during acquisition. The NCO outputs start at the specified phase values when the start trigger is received.

- Data Processing Mode - [ScopeDdcDataProcessingMode](#)

The Data Processing Mode Selection stage selects between [Real](#) and [Complex](#) processing. In Real mode, only the Real I data is returned when a waveform fetch is executed. In Complex mode, the I and Q data streams are interleaved before they are stored to memory. When the complex data is fetched, the data is returned as interleaved I and Q data. The data can be fetched as an array of complex IQ samples, as an array of interleaved real I and Q samples, or as separate arrays of real I and Q samples.

Note

Additional properties are available for instruments that support OSP. You access these properties, [DdcDataProcessingMode](#) and [OverflowErrorReportingMode](#), from the [ScopeAcquisition](#) class. [DdcDataProcessingMode](#) and [OverflowErrorReportingMode](#) are not channel specific.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)



[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Properties

The [ScopeChannelOnboardSignalProcessing](#) type exposes the following members.

Properties

Name	Description
 Ddc	Gets the ScopeChannelDdc sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.
 EqualizationFilter	Gets the ScopeChannelEqualizationFilter sub-object that is related to configuring the Equalization Filter onboard the high-speed digitizer.

[Top](#)

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingDdc Property

Gets the [ScopeChannelDdc](#) sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelDdc Ddc { get; }
```

```
Public ReadOnly Property Ddc As ScopeChannelDdc  
    Get
```

Property Value

Type: [ScopeChannelDdc](#)

An object of type [ScopeChannelDdc](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingDdc Property

Gets the [ScopeChannelDdc](#) sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelDdc Ddc { get; }
```

```
Public ReadOnly Property Ddc As ScopeChannelDdc  
    Get
```

Property Value

Type: [ScopeChannelDdc](#)

An object of type [ScopeChannelDdc](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingEqualizationFilter

Gets the [ScopeChannelEqualizationFilter](#) sub-object that is related to configuring the Equalization Filter onboard the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelEqualizationFilter EqualizationFilter { get; }
```

```
Public ReadOnly Property EqualizationFilter As ScopeChannelEqualizationFilter  
    Get
```

Property Value

Type: [ScopeChannelEqualizationFilter](#)

An object of type [ScopeChannelEqualizationFilter](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Properties

The [ScopeChannelOnboardSignalProcessing](#) type exposes the following members.

Properties

Name	Description
 Ddc	Gets the ScopeChannelDdc sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.
 EqualizationFilter	Gets the ScopeChannelEqualizationFilter sub-object that is related to configuring the Equalization Filter onboard the high-speed digitizer.

[Top](#)

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingDdc Property

Gets the [ScopeChannelDdc](#) sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelDdc Ddc { get; }
```



```
Public ReadOnly Property Ddc As ScopeChannelDdc
    Get
```

Property Value

Type: [ScopeChannelDdc](#)

An object of type [ScopeChannelDdc](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingDdc Property

Gets the [ScopeChannelDdc](#) sub-object that is related to configuring the digital downconverter (DDC) block of the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelDdc Ddc { get; }
```

```
Public ReadOnly Property Ddc As ScopeChannelDdc
    Get
```

Property Value

Type: [ScopeChannelDdc](#)

An object of type [ScopeChannelDdc](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessingEqualizationFilter Property

Gets the [ScopeChannelEqualizationFilter](#) sub-object that is related to configuring the Equalization Filter onboard the high-speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeChannelEqualizationFilter EqualizationFilter { get; }
```

```
Public ReadOnly Property EqualizationFilter As ScopeChannelEqualizationFilter  
    Get
```

Property Value

Type: [ScopeChannelEqualizationFilter](#)

An object of type [ScopeChannelEqualizationFilter](#).

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142.

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelOnboardSignalProcessing Methods

The [ScopeChannelOnboardSignalProcessing](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeChannelOnboardSignalProcessing Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeChannelTerminalConfiguration Enumeration

Specifies the terminal configuration for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeChannelTerminalConfiguration
```

```
Public Enumeration ScopeChannelTerminalConfiguration
```

Members

Member name	Value	Description
SingleEnded	0	The channel is single-ended.
UnbalancedDifferential	1	The channel is unbalanced differential.
Differential	2	The channel is differential.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Class

Represents the line on which the sample clock is sent or received.

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax













C#VB

```
public sealed class ScopeClockSynchronizationPulseSource
```

```
Public NotInheritable Class ScopeClockSynchronizationPulseSource
```


The ScopeClockSynchronizationPulseSource type exposes the following members.







Properties

	Name	Description
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.

Top









Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified object are equal.

		(Overrides ObjectEquals(Object).)
	Equals(ScopeClockSynchronizationPulseSource)	Determines whether the current instance of <code>ScopeClockSynchronizationPulseSource</code> and the specified <code>ScopeClockSynchronizationPulseSource</code> object are equal.
 	FromString	Creates a <code>ScopeClockSynchronizationPulseSource</code> object from the specified String .
	GetHashCode	Gets the hash code for the current instance of <code>ScopeClockSynchronizationPulseSource</code> . (Overrides ObjectGetHashCode.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Converts the current instance of <code>ScopeClockSynchronizationPulseSource</code> to String . (Overrides ObjectToString.)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two <code>ScopeClockSynchronizationPulseSource</code> object instances are equal.
 	(String to ScopeClockSynchronizationPulseSource)	Converts a specified String to an equivalent <code>ScopeClockSynchronizationPulseSource</code> object.
 	(ScopeClockSynchronizationPulseSource to String)	Converts a <code>ScopeClockSynchronizationPulseSource</code> object to an equivalent String .
 	Inequality	Checks whether two <code>ScopeClockSynchronizationPulseSource</code> object instances are not equal.

[Top](#)

Remarks

This line should be the same for all devices to be synchronized. For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference











[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Properties

The [ScopeClockSynchronizationPulseSource](#) type exposes the following members.

Properties

	Name	Description
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.

 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceNoSource P

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static ScopeClockSynchronizationPulseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)













[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Properties

The [ScopeClockSynchronizationPulseSource](#) type exposes the following members.

Properties

	Name	Description
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeClockSynchronizationPulseSource  
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeClockSynchronizationPulseSource
    Get
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

A [ScopeClockSynchronizationPulseSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Methods

The [ScopeClockSynchronizationPulseSource](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeClockSynchronizationPulseSource)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified ScopeClockSynchronizationPulseSource object are equal.
 FromString 	Creates a ScopeClockSynchronizationPulseSource object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeClockSynchronizationPulseSource . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeClockSynchronizationPulseSource to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Meth

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeClockSynchronizationPulseSource)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified ScopeClockSynchronizationPulseSource object are equal.

[Top](#)

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Meth

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Meth

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)
[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Method

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified [ScopeClockSynchronizationPulseSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeClockSynchronizationPulseSource source  
)
```

```
Public Function Equals (  
    source As ScopeClockSynchronizationPulseSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the [ScopeClockSynchronizationPulseSource](#) object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeClockSynchronizationPulseSource)	Determines whether the current instance of ScopeClockSynchronizationPulseSource and the specified ScopeClockSynchronizationPulseSource object are equal.

[Top](#)

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Meth

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Method (Object)

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquals Method (ScopeClockSynchronizationPulseSource)

Determines whether the current instance of [ScopeClockSynchronizationPulseSource](#) and the specified [ScopeClockSynchronizationPulseSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeClockSynchronizationPulseSource source  
)
```

```
Public Function Equals (  
    source As ScopeClockSynchronizationPulseSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the [ScopeClockSynchronizationPulseSource](#) object to compare to the current instance of [ScopeClockSynchronizationPulseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceFromString Method

Creates a [ScopeClockSynchronizationPulseSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeClockSynchronizationPulseSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeClockSynchronizationPulseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeClockSynchronizationPulseSource](#) object returned represents.

Return Value

Type: [ScopeClockSynchronizationPulseSource](#)

Returns an object of type [ScopeClockSynchronizationPulseSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceGetHashCode Method

Gets the hash code for the current instance of [ScopeClockSynchronizationPulseSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeClockSynchronizationPulseSource](#).

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceToString Method

Converts the current instance of [ScopeClockSynchronizationPulseSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeClockSynchronizationPulseSource](#).

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Operators and Type Conversions

The [ScopeClockSynchronizationPulseSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeClockSynchronizationPulseSource object instances are equal.
	(String to ScopeClockSynchronizationPulseSource)	Converts a specified String to an equivalent ScopeClockSynchronizationPulseSource object.
	(ScopeClockSynchronizationPulseSource to String)	Converts a ScopeClockSynchronizationPulseSource object to an equivalent String .
	Inequality	Checks whether two ScopeClockSynchronizationPulseSource object instances are not equal.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquality Operator

Checks whether two [ScopeClockSynchronizationPulseSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeClockSynchronizationPulseSource source1,
    ScopeClockSynchronizationPulseSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeClockSynchronizationPulseSource,
    source2 As ScopeClockSynchronizationPulseSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
 Specifies the first [ScopeClockSynchronizationPulseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
 Specifies the second [ScopeClockSynchronizationPulseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceEquality Operator

Checks whether two [ScopeClockSynchronizationPulseSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeClockSynchronizationPulseSource source1,
    ScopeClockSynchronizationPulseSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeClockSynchronizationPulseSource,
    source2 As ScopeClockSynchronizationPulseSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the first [ScopeClockSynchronizationPulseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the second [ScopeClockSynchronizationPulseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference





[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Conversion Operators

Overload List

	Name	Description
 	(String to ScopeClockSynchronizationPulseSource)	Converts a specified String to an equivalent ScopeClockSynchronizationPulseSource object.
 	(ScopeClockSynchronizationPulseSource to String)	Converts a ScopeClockSynchronizationPulseSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Conversion (String to ScopeClockSynchronizationPulseSource)

Converts a specified [String](#) to an equivalent [ScopeClockSynchronizationPulseSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeClockSynchronizationPulseSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeClockSynchronizationPulseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeClockSynchronizationPulseSource](#) object.

Return Value

Type: [ScopeClockSynchronizationPulseSource](#)

Returns the [ScopeClockSynchronizationPulseSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Conversion (String to ScopeClockSynchronizationPulseSource)

Converts a specified [String](#) to an equivalent [ScopeClockSynchronizationPulseSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeClockSynchronizationPulseSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeClockSynchronizationPulseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeClockSynchronizationPulseSource](#) object.

Return Value

Type: [ScopeClockSynchronizationPulseSource](#)

Returns the [ScopeClockSynchronizationPulseSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSource Conversion
(ScopeClockSynchronizationPulseSource to String)

Converts a [ScopeClockSynchronizationPulseSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeClockSynchronizationPulseSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeClockSynchronizationPulseSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the [ScopeClockSynchronizationPulseSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeClockSynchronizationPulseSource](#) object.

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeClockSynchronizationPulseSourceInequality Operator

Checks whether two [ScopeClockSynchronizationPulseSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static bool operator !=(  
    ScopeClockSynchronizationPulseSource source1,  
    ScopeClockSynchronizationPulseSource source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeClockSynchronizationPulseSource,  
    source2 As ScopeClockSynchronizationPulseSource  
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the first [ScopeClockSynchronizationPulseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
Specifies the second [ScopeClockSynchronizationPulseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeClockSynchronizationPulseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo Structure

Specifies an array of structures containing gain and offset coefficients for a given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public struct ScopeCoefficientInfo
```

```
Public Structure ScopeCoefficientInfo
```






The ScopeCoefficientInfo type exposes the following members.

Properties

	Name	Description
	Gain	Gets the gain factor of the given channel.
	Offset	Gets the offset factor of the given channel.



[Top](#)

Methods

	Name	Description
	Equals(Object)	Determines if the current instance of ScopeCoefficientInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
	Equals(ScopeCoefficientInfo)	Determines whether the current instance of ScopeCoefficientInfo and the ScopeCoefficientInfo object that you specify are equal.
	GetHashCode	Returns the hash code for the current instance of ScopeCoefficientInfo. (Overrides ValueTypeGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether the two instances of ScopeCoefficientInfo are equal.
	Inequality	Checks whether the two instances of ScopeCoefficientInfo are unequal.

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo Properties

The [ScopeCoefficientInfo](#) type exposes the following members.

Properties

	Name	Description
	Gain	Gets the gain factor of the given channel.
	Offset	Gets the offset factor of the given channel.

[Top](#)

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double  
    Get
```

Property Value

Type: [Double](#)

Gain factor.

Remarks

Gain is used for normalizing binary data using the formula $\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double  
    Get
```

Property Value

Type: [Double](#)

Gain factor.

Remarks

Gain is used for normalizing binary data using the formula $\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoOffset Property

Gets the offset factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Offset { get; }
```

```
Public ReadOnly Property Offset As Double
    Get
```

Property Value

Type: [Double](#)

Offset factor.

Remarks

Offset is used for normalizing binary data using the formula normalized binary data = raw binary data × gain factor + offset

See Also

Reference

[ScopeCoefficientInfo Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo Properties

The [ScopeCoefficientInfo](#) type exposes the following members.

Properties

	Name	Description
	Gain	Gets the gain factor of the given channel.
	Offset	Gets the offset factor of the given channel.

[Top](#)

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double  
    Get
```

Property Value

Type: [Double](#)

Gain factor.

Remarks

Gain is used for normalizing binary data using the formula $\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$

See Also

Reference

[ScopeCoefficientInfo Structure](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double  
    Get
```

Property Value

Type: [Double](#)

Gain factor.

Remarks

Gain is used for normalizing binary data using the formula $\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoOffset Property

Gets the offset factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Offset { get; }
```

```
Public ReadOnly Property Offset As Double  
    Get
```

Property Value

Type: [Double](#)

Offset factor.

Remarks

Offset is used for normalizing binary data using the formula $\text{normalized binary data} = \text{raw binary data} \times \text{gain factor} + \text{offset}$

See Also

Reference

[ScopeCoefficientInfo Structure](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo Methods

The [ScopeCoefficientInfo](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines if the current instance of ScopeCoefficientInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeCoefficientInfo)	Determines whether the current instance of ScopeCoefficientInfo and the ScopeCoefficientInfo object that you specify are equal.
 GetHashCode	Returns the hash code for the current instance of ScopeCoefficientInfo . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

See Also

Reference


[ScopeCoefficientInfo Structure](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method

Overload List

Name	Description
 Equals(Object)	Determines if the current instance of ScopeCoefficientInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)

	Equals(ScopeCoefficientInfo)	Determines whether the current instance of ScopeCoefficientInfo and the ScopeCoefficientInfo object that you specify are equal.
---	--	---

[Top](#)

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (Object)

Determines if the current instance of [ScopeCoefficientInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (Object)

Determines if the current instance of [ScopeCoefficientInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (ScopeCoefficientInfo)

Determines whether the current instance of [ScopeCoefficientInfo](#) and the [ScopeCoefficientInfo](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeCoefficientInfo coefficientInfo  
)
```

```
Public Function Equals (
    coefficientInfo As ScopeCoefficientInfo
) As Boolean
```

Parameters

coefficientInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
 The [ScopeCoefficientInfo](#) object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method

Overload List

Name	Description
 Equals(Object)	Determines if the current instance of ScopeCoefficientInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeCoefficientInfo)	Determines whether the current instance of ScopeCoefficientInfo and the ScopeCoefficientInfo object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (Object)

Determines if the current instance of [ScopeCoefficientInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (Object)

Determines if the current instance of [ScopeCoefficientInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquals Method (ScopeCoefficientInfo)

Determines whether the current instance of [ScopeCoefficientInfo](#) and the [ScopeCoefficientInfo](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeCoefficientInfo coefficientInfo  
)
```

```
Public Function Equals (
    coefficientInfo As ScopeCoefficientInfo
) As Boolean
```

Parameters

coefficientInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
 The [ScopeCoefficientInfo](#) object to be compared to the current instance of [ScopeCoefficientInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo.GetHashCode Method

Returns the hash code for the current instance of [ScopeCoefficientInfo](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) that represents the hash code for the current instance of [ScopeCoefficientInfo](#).

See Also

Reference

[ScopeCoefficientInfo Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfo Operators

The [ScopeCoefficientInfo](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether the two instances of ScopeCoefficientInfo are equal.
	Inequality	Checks whether the two instances of ScopeCoefficientInfo are unequal.

[Top](#)

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquality Operator

Checks whether the two instances of [ScopeCoefficientInfo](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeCoefficientInfo coefficientInfo1,
    ScopeCoefficientInfo coefficientInfo2
)
```

```
Public Shared Operator = (
    coefficientInfo1 As ScopeCoefficientInfo,
    coefficientInfo2 As ScopeCoefficientInfo
) As Boolean
```

Parameters

coefficientInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
A [ScopeCoefficientInfo](#) object.

coefficientInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
A [ScopeCoefficientInfo](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoEquality Operator

Checks whether the two instances of [ScopeCoefficientInfo](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeCoefficientInfo coefficientInfo1,
    ScopeCoefficientInfo coefficientInfo2
)
```

```
Public Shared Operator = (
    coefficientInfo1 As ScopeCoefficientInfo,
    coefficientInfo2 As ScopeCoefficientInfo
) As Boolean
```

Parameters

coefficientInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
A [ScopeCoefficientInfo](#) object.

coefficientInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
A [ScopeCoefficientInfo](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoefficientInfoInequality Operator

Checks whether the two instances of [ScopeCoefficientInfo](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeCoefficientInfo coefficientInfo1,
    ScopeCoefficientInfo coefficientInfo2
)
```

```
Public Shared Operator <> (
    coefficientInfo1 As ScopeCoefficientInfo,
    coefficientInfo2 As ScopeCoefficientInfo
) As Boolean
```

Parameters

coefficientInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)

A [ScopeCoefficientInfo](#) object.

coefficientInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeCoefficientInfo](#)
 A [ScopeCoefficientInfo](#) object.

Return Value

Type: [Boolean](#)
 if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeCoefficientInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgs Class

Represents the event data obtained when a coercion event occurs.

Inheritance Hierarchy

[SystemObject](#)

[SystemEventArgs](#)

NationalInstruments.ModularInstruments.NIScopeScopeCoercionEventArgs

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeCoercionEventArgs : EventArgs
```

```
Public NotInheritable Class ScopeCoercionEventArgs
    Inherits EventArgs
```






The ScopeCoercionEventArgs type exposes the following members.

Properties

	Name	Description
	Text	Gets the coercion message.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgs Properties

The [ScopeCoercionEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Text	Gets the coercion message.

[Top](#)

See Also

Reference

[ScopeCoercionEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgsText Property

Gets the coercion message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that represents the coercion message.

See Also

Reference

[ScopeCoercionEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgsText Property

Gets the coercion message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that represents the coercion message.

See Also

Reference

[ScopeCoercionEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgs Properties

The [ScopeCoercionEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Text	Gets the coercion message.

[Top](#)

See Also

Reference

[ScopeCoercionEventArgs Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgsText Property

Gets the coercion message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that represents the coercion message.

See Also

Reference

[ScopeCoercionEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgsText Property

Gets the coercion message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that represents the coercion message.

See Also

Reference

[ScopeCoercionEventArgs Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeCoercionEventArgs Methods

The [ScopeCoercionEventArgs](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeCoercionEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDdcDataProcessingMode Enumeration

Specifies how the digital downconverter (DDC) block of a high-speed digitizer should process the data.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeDdcDataProcessingMode
```

```
Public Enumeration ScopeDdcDataProcessingMode
```

Members

Member name	Value	Description
Real	0	The DDC block processes only the real part, I data, of the waveform data points.
Complex	1	The DDC block processes both the real, I, and imaginary, Q, parts of the waveform data points.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[ScopeAcquisitionDdcDataProcessingMode](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTrigger Class

Represents the configuration parameters for the digital-edge Advance trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeAdvanceTrigger

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeDigitalEdgeAdvanceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeDigitalEdgeAdvanceTrigger
    Inherits ScopeSubObject
```





The `ScopeDigitalEdgeAdvanceTrigger` type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Configure Trigger topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTrigger Properties

The [ScopeDigitalEdgeAdvanceTrigger](#) type exposes the following members.

Properties

	Name	Description
	Source	Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTriggerSource Property

Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeAdvanceTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

An object of type [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTriggerSource Property

Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeAdvanceTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

An object of type [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTrigger Properties

The [ScopeDigitalEdgeAdvanceTrigger](#) type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTriggerSource Property

Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeAdvanceTriggerSource
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

An object of type [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTriggerSource Property

Gets or sets the source the digitizer monitors for an advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeAdvanceTriggerSource
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerSource](#)

An object of type [ScopeAdvanceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeAdvanceTrigger Methods

The [ScopeDigitalEdgeAdvanceTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDigitalEdgeAdvanceTrigger Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTrigger Class

Represents the configuration parameters for the digital-edge arm Reference trigger, such as the trigger source and the amount of trigger delay.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeArmReferenceTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeDigitalEdgeArmReferenceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeDigitalEdgeArmReferenceTrigger
    Inherits ScopeSubObject
```





The ScopeDigitalEdgeArmReferenceTrigger type exposes the following members.

Properties

Name	Description
 <u>Source</u>	Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

Top

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Configure Trigger topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTrigger Properties

The [ScopeDigitalEdgeArmReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTriggerSource Property

Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeArmReferenceTriggerSource
    Get
    Set
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

An object of type [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTriggerSource Property

Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeArmReferenceTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

An object of type [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTrigger Properties

The [ScopeDigitalEdgeArmReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTriggerSource Property

Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeArmReferenceTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

An object of type [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTriggerSource Property

Gets or sets the source the digitizer monitors for an arm reference trigger. When the arm reference trigger is received, the digitizer begins looking for a Reference (Stop) trigger from the configured trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeArmReferenceTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeArmReferenceTriggerSource](#)

An object of type [ScopeArmReferenceTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeArmReferenceTrigger Methods

The [ScopeDigitalEdgeArmReferenceTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDigitalEdgeArmReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTrigger Class

Represents the [ScopeDigitalEdgeStartTrigger](#) sub-object that contains properties and methods related to the digital-edge Start trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTrigger](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax


C#VB

```
public sealed class ScopeDigitalEdgeStartTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeDigitalEdgeStartTrigger
    Inherits ScopeSubObject
```





The `ScopeDigitalEdgeStartTrigger` type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Configure Trigger topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTrigger Properties

The [ScopeDigitalEdgeStartTrigger](#) type exposes the following members.

Properties

	Name	Description
	Source	Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Property

Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeDigitalEdgeStartTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

An object of type [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Property

Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeDigitalEdgeStartTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

An object of type [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTrigger Properties

The [ScopeDigitalEdgeStartTrigger](#) type exposes the following members.

Properties

Name	Description
 Source	Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Property

Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeDigitalEdgeStartTriggerSource
    Get
    Set
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

An object of type [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Property

Gets or sets the source the digitizer monitors for an acquisition arm (Start) trigger. When the acquisition arm trigger is received, the digitizer begins acquiring pretrigger samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeDigitalEdgeStartTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

An object of type [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTrigger Methods

The [ScopeDigitalEdgeStartTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTrigger Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Class

Represents the source terminal for [ScopeDigitalEdgeStartTrigger](#).

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax









C#VB







```
public sealed class ScopeDigitalEdgeStartTriggerSource
```

```
Public NotInheritable Class ScopeDigitalEdgeStartTriggerSource
```

The ScopeDigitalEdgeStartTriggerSource type exposes the following members.








Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.

 Rtsi3	Gets the source terminal for RTSI 3.
 Rtsi4	Gets the source terminal for RTSI 4.
 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.



[Top](#)




Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of <code>ScopeDigitalEdgeStartTriggerSource</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeDigitalEdgeStartTriggerSource)	Determines whether the current instance of <code>ScopeDigitalEdgeStartTriggerSource</code> and the specified <code>ScopeDigitalEdgeStartTriggerSource</code> object are equal.
 	FromString	Creates a <code>ScopeDigitalEdgeStartTriggerSource</code> object from the specified String .
	GetHashCode	Gets the hash code for the current instance of <code>ScopeDigitalEdgeStartTriggerSource</code> . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of <code>ScopeDigitalEdgeStartTriggerSource</code> to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two <code>ScopeDigitalEdgeStartTriggerSource</code> object instances are equal.

 S (String to ScopeDigitalEdgeStartTriggerSource)	Converts a specified String to an equivalent ScopeDigitalEdgeStartTriggerSource object.
 S (ScopeDigitalEdgeStartTriggerSource to String)	Converts a ScopeDigitalEdgeStartTriggerSource object to an equivalent String .
 S Inequality	Checks whether two ScopeDigitalEdgeStartTriggerSource object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerScopeDigitalEdgeStartTriggerSource](#)














Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Properties

The [ScopeDigitalEdgeStartTriggerSource](#) type exposes the following members.

Properties

	Name	Description
 S	Immediate	Gets the source terminal for Immediate.

 Pfi0	Gets the source terminal for PFI 0.
 Pfi1	Gets the source terminal for PFI 1.
 Pfi2	Gets the source terminal for PFI 2.
 PxiStar	Gets the source terminal for PXI STAR.
 Rtsi0	Gets the source terminal for RTSI 0.
 Rtsi1	Gets the source terminal for RTSI 1.
 Rtsi2	Gets the source terminal for RTSI 2.
 Rtsi3	Gets the source terminal for RTSI 3.
 Rtsi4	Gets the source terminal for RTSI 4.
 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static ScopeDigitalEdgeStartTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtSi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceSoftwareTriggerFunction

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource SoftwareTriggerFunction
{ get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As
ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)















[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Properties

The [ScopeDigitalEdgeStartTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Immediate	Gets the source terminal for Immediate.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiStar	Gets the source terminal for PXI STAR.
	Rtsi0	Gets the source terminal for RTSI 0.
	Rtsi1	Gets the source terminal for RTSI 1.
	Rtsi2	Gets the source terminal for RTSI 2.
	Rtsi3	Gets the source terminal for RTSI 3.
	Rtsi4	Gets the source terminal for RTSI 4.
	Rtsi5	Gets the source terminal for RTSI 5.
	Rtsi6	Gets the source terminal for RTSI 6.
	Rtsi7	Gets the source terminal for RTSI 7.
	SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

[See Also](#)

[Reference](#)

[ScopeDigitalEdgeStartTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeDigitalEdgeStartTriggerSource  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi6 { get; }
```



```
Public Shared ReadOnly Property Rtsi6 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceSoftwareTriggerFunction Property

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource SoftwareTriggerFunction
{ get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As
ScopeDigitalEdgeStartTriggerSource
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

A [ScopeDigitalEdgeStartTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Methods

The [ScopeDigitalEdgeStartTriggerSource](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeDigitalEdgeStartTriggerSource)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified ScopeDigitalEdgeStartTriggerSource object are equal.
 FromString 	Creates a ScopeDigitalEdgeStartTriggerSource object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeDigitalEdgeStartTriggerSource . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeDigitalEdgeStartTriggerSource to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeDigitalEdgeStartTriggerSource)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified ScopeDigitalEdgeStartTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (ScopeDigitalEdgeStartTriggerSource)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified [ScopeDigitalEdgeStartTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeDigitalEdgeStartTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeDigitalEdgeStartTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the [ScopeDigitalEdgeStartTriggerSource](#) object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeDigitalEdgeStartTriggerSource)	Determines whether the current instance of ScopeDigitalEdgeStartTriggerSource and the specified ScopeDigitalEdgeStartTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquals Method (ScopeDigitalEdgeStartTriggerSource)

Determines whether the current instance of [ScopeDigitalEdgeStartTriggerSource](#) and the specified [ScopeDigitalEdgeStartTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeDigitalEdgeStartTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeDigitalEdgeStartTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the [ScopeDigitalEdgeStartTriggerSource](#) object to compare to the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceFromString Method

Creates a [ScopeDigitalEdgeStartTriggerSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeDigitalEdgeStartTriggerSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeDigitalEdgeStartTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeDigitalEdgeStartTriggerSource](#) object returned represents.

Return Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

Returns an object of type [ScopeDigitalEdgeStartTriggerSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceToString Method

Converts the current instance of [ScopeDigitalEdgeStartTriggerSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeDigitalEdgeStartTriggerSource](#).

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Operators and Type Conversions

The [ScopeDigitalEdgeStartTriggerSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeDigitalEdgeStartTriggerSource object instances are equal.
	(String to ScopeDigitalEdgeStartTriggerSource)	Converts a specified String to an equivalent ScopeDigitalEdgeStartTriggerSource object.
	(ScopeDigitalEdgeStartTriggerSource to String)	Converts a ScopeDigitalEdgeStartTriggerSource object to an equivalent String .
	Inequality	Checks whether two ScopeDigitalEdgeStartTriggerSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquality Operator

Checks whether two [ScopeDigitalEdgeStartTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeDigitalEdgeStartTriggerSource source1,
    ScopeDigitalEdgeStartTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeDigitalEdgeStartTriggerSource,
    source2 As ScopeDigitalEdgeStartTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the first [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the second [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceEquality Operator

Checks whether two [ScopeDigitalEdgeStartTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeDigitalEdgeStartTriggerSource source1,
    ScopeDigitalEdgeStartTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeDigitalEdgeStartTriggerSource,
    source2 As ScopeDigitalEdgeStartTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the first [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the second [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeDigitalEdgeStartTriggerSource)	Converts a specified String to an equivalent ScopeDigitalEdgeStartTriggerSource object.
	(ScopeDigitalEdgeStartTriggerSource to String)	Converts a ScopeDigitalEdgeStartTriggerSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Conversion (String to ScopeDigitalEdgeStartTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeDigitalEdgeStartTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeDigitalEdgeStartTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeDigitalEdgeStartTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeDigitalEdgeStartTriggerSource](#) object.

Return Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

Returns the [ScopeDigitalEdgeStartTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Conversion (String to ScopeDigitalEdgeStartTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeDigitalEdgeStartTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeDigitalEdgeStartTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeDigitalEdgeStartTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeDigitalEdgeStartTriggerSource](#) object.

Return Value

Type: [ScopeDigitalEdgeStartTriggerSource](#)

Returns the [ScopeDigitalEdgeStartTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

Overload

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSource Conversion
(ScopeDigitalEdgeStartTriggerSource to String)

Converts a [ScopeDigitalEdgeStartTriggerSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (
    ScopeDigitalEdgeStartTriggerSource source
)
```

```
Public Shared Widening Operator CType (
    source As ScopeDigitalEdgeStartTriggerSource
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the [ScopeDigitalEdgeStartTriggerSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeDigitalEdgeStartTriggerSource](#) object.

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDigitalEdgeStartTriggerSourceInequality Operator

Checks whether two [ScopeDigitalEdgeStartTriggerSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeDigitalEdgeStartTriggerSource source1,
    ScopeDigitalEdgeStartTriggerSource source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeDigitalEdgeStartTriggerSource,
    source2 As ScopeDigitalEdgeStartTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
Specifies the first [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTriggerSource](#)
 Specifies the second [ScopeDigitalEdgeStartTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeDigitalEdgeStartTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentity Class

Provides the methods and properties that provide information about the instrument and the NI-SCOPE driver.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeDriverIdentity](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax











C#VB

```
public sealed class ScopeDriverIdentity : ScopeSubObject,
    IIviDriverIdentity, IIviComponentIdentity
```

```
Public NotInheritable Class ScopeDriverIdentity
    Inherits ScopeSubObject
    Implements IIviDriverIdentity, IIviComponentIdentity
```


The ScopeDriverIdentity type exposes the following members.






Properties

Name	Description
 Description	Gets a string that contains a brief description of NI-SCOPE driver.
 Identifier	Gets a string that contains an identifier for NI-SCOPE.NET API.
 InstrumentFirmwareRevision	Gets the firmware revision information for the NI-SCOPE instrument currently in use.
 InstrumentManufacturer	Gets the name of the manufacturer for the NI-SCOPE instrument currently in use.
 InstrumentModel	Gets the model number or name of the NI-SCOPE instrument currently in use.
 Revision	Gets a string that contains additional version information about the NI-SCOPE driver.
 SerialNumber	Gets the serial number of the instrument.
 SpecificationMajorVersion	Gets an integer that specifies the major version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 SpecificationMinorVersion	Gets a integer that specifies the minor version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 Vendor	Gets a string that contains the name of the vendor that supplies the NI-SCOPE driver.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)

 GetGroupCapabilities	Returns a list of names of class capability groups that the IVI specific driver implements.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetSupportedInstrumentModels	Returns a list of names of instrument models with which the IVI specific driver is compatible.
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to [NI High Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)










Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentity Properties

The [ScopeDriverIdentity](#) type exposes the following members.

Properties

Name	Description
 Description	Gets a string that contains a brief description of NI-SCOPE driver.

 Identifier	Gets a string that contains an identifier for NI-SCOPE.NET API.
 InstrumentFirmwareRevision	Gets the firmware revision information for the NI-SCOPE instrument currently in use.
 InstrumentManufacturer	Gets the name of the manufacturer for the NI-SCOPE instrument currently in use.
 InstrumentModel	Gets the model number or name of the NI-SCOPE instrument currently in use.
 Revision	Gets a string that contains additional version information about the NI-SCOPE driver.
 SerialNumber	Gets the serial number of the instrument.
 SpecificationMajorVersion	Gets an integer that specifies the major version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 SpecificationMinorVersion	Gets a integer that specifies the minor version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 Vendor	Gets a string that contains the name of the vendor that supplies the NI-SCOPE driver.

[Top](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityDescription Property

Gets a string that contains a brief description of NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Description { get; }
```

```
Public ReadOnly Property Description As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains a brief description about NI-SCOPE driver.

Implements

[IIVIComponentIdentityDescription](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityDescription Property

Gets a string that contains a brief description of NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Description { get; }
```

```
Public ReadOnly Property Description As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains a brief description about NI-SCOPE driver.

Implements

[IIVIComponentIdentityDescription](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityIdentifier Property

Gets a string that contains an identifier for NI-SCOPE.NET API.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Identifier { get; }
```

```
Public ReadOnly Property Identifier As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains an identifier for NI-SCOPE.NET API.

Implements

[IIVIInstrumentIdentifier](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentFirmwareRevision Property

Gets the firmware revision information for the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string InstrumentFirmwareRevision { get; }
```

```
Public ReadOnly Property InstrumentFirmwareRevision As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the firmware revision information for the NI-SCOPE instrument in use.

Implements

IlviDriverIdentityInstrumentFirmwareRevision

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentManufacturer Property

Gets the name of the manufacturer for the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/**VB**

```
public string InstrumentManufacturer { get; }
```

```
Public ReadOnly Property InstrumentManufacturer As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the name of the manufacturer for the NI-SCOPE instrument currently in use.

Implements

IlviDriverIdentityInstrumentManufacturer

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentModel Property

Gets the model number or name of the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string InstrumentModel { get; }
```

```
Public ReadOnly Property InstrumentModel As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the model number or name of the NI-SCOPE instrument currently in use.

Implements

[IIVIIdentityInstrumentModel](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityRevision Property

Gets a string that contains additional version information about the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Revision { get; }
```

```
Public ReadOnly Property Revision As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains additional version information about the NI-SCOPE driver.

Implements

[IIVIComponentIdentityRevision](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySerialNumber Property

Gets the serial number of the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string SerialNumber { get; }
```

```
Public ReadOnly Property SerialNumber As String
    Get
```

Property Value

Type: [String](#)

The serial number of the instrument.

Exceptions

Exception	Condition
ObjectDisposedException	SerialNumber was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySpecificationMajorVersion Property

Gets an integer that specifies the major version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int SpecificationMajorVersion { get; }
```

```
Public ReadOnly Property SpecificationMajorVersion As Integer  
    Get
```

Property Value

Type: [Int32](#)

A [Int32](#) that contains the major version number of the NI-SCOPE .NET class library.

Implements

[IIVIIdentitySpecificationMajorVersion](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySpecificationMinorVersion Property

Gets a integer that specifies the minor version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int SpecificationMinorVersion { get; }
```

```
Public ReadOnly Property SpecificationMinorVersion As Integer  
    Get
```

Property Value

Type: [Int32](#)

A [Int32](#) that contains the minor version number of the NI-SCOPE .NET class library.

Implements

[IIVIIdentitySpecificationMinorVersion](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityVendor Property

Gets a string that contains the name of the vendor that supplies the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Vendor { get; }
```

```
Public ReadOnly Property Vendor As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the name of the vendor that supplies the NI-SCOPE driver.

Implements

[IIVIComponentIdentityVendor](#)

See Also

Reference

[ScopeDriverIdentity Class](#)











[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentity Properties

The [ScopeDriverIdentity](#) type exposes the following members.

Properties

Name	Description
 Description	Gets a string that contains a brief description of NI-SCOPE driver.
 Identifier	Gets a string that contains an identifier for NI-SCOPE.NET API.
 InstrumentFirmwareRevision	Gets the firmware revision information for the NI-SCOPE instrument currently in use.
 InstrumentManufacturer	Gets the name of the manufacturer for the NI-SCOPE instrument currently in use.
 InstrumentModel	Gets the model number or name of the NI-SCOPE instrument currently in use.
 Revision	Gets a string that contains additional version information about the NI-SCOPE driver.
 SerialNumber	Gets the serial number of the instrument.
 SpecificationMajorVersion	Gets an integer that specifies the major version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 SpecificationMinorVersion	Gets a integer that specifies the minor version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.
 Vendor	Gets a string that contains the name of the vendor that supplies the NI-SCOPE driver.

[Top](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityDescription Property

Gets a string that contains a brief description of NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Description { get; }
```

```
Public ReadOnly Property Description As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains a brief description about NI-SCOPE driver.

Implements

[IIVIComponentIdentityDescription](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityDescription Property

Gets a string that contains a brief description of NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Description { get; }
```

```
Public ReadOnly Property Description As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains a brief description about NI-SCOPE driver.

Implements

[IIVIComponentIdentityDescription](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityIdentifier Property

Gets a string that contains an identifier for NI-SCOPE.NET API.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public string Identifier { get; }
```

```
Public ReadOnly Property Identifier As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains an identifier for NI-SCOPE.NET API.

Implements

[IIVIIdentityIdentifier](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentFirmwareRevision Property

Gets the firmware revision information for the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public string InstrumentFirmwareRevision { get; }
```

```
Public ReadOnly Property InstrumentFirmwareRevision As String
    Get
```


Property Value

Type: [String](#)

A [String](#) that contains the firmware revision information for the NI-SCOPE instrument in use.

Implements

[IIVIInstrumentFirmwareRevision](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentManufacturer Property

Gets the name of the manufacturer for the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string InstrumentManufacturer { get; }
```

```
Public ReadOnly Property InstrumentManufacturer As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the name of the manufacturer for the NI-SCOPE instrument currently in use.

Implements

IlviDriverIdentityInstrumentManufacturer

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityInstrumentModel Property

Gets the model number or name of the NI-SCOPE instrument currently in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string InstrumentModel { get; }
```

```
Public ReadOnly Property InstrumentModel As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the model number or name of the NI-SCOPE instrument currently in use.

Implements

IlviDriverIdentityInstrumentModel

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityRevision Property

Gets a string that contains additional version information about the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Revision { get; }
```

```
Public ReadOnly Property Revision As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains additional version information about the NI-SCOPE driver.

Implements

[IIVIComponentIdentityRevision](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySerialNumber Property

Gets the serial number of the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string SerialNumber { get; }
```

```
Public ReadOnly Property SerialNumber As String
    Get
```

Property Value

Type: [String](#)

The serial number of the instrument.

Exceptions

Exception	Condition
ObjectDisposedException	SerialNumber was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySpecificationMajorVersion Property

Gets an integer that specifies the major version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int SpecificationMajorVersion { get; }
```

```
Public ReadOnly Property SpecificationMajorVersion As Integer  
    Get
```

Property Value

Type: [Int32](#)

A [Int32](#) that contains the major version number of the NI-SCOPE .NET class library.

Implements

[IIVIIdentitySpecificationMajorVersion](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentitySpecificationMinorVersion Property

Gets a integer that specifies the minor version number of the class specification in accordance with which the NI-SCOPE .NET class library was developed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int SpecificationMinorVersion { get; }
```

```
Public ReadOnly Property SpecificationMinorVersion As Integer  
    Get
```

Property Value

Type: [Int32](#)

A [Int32](#) that contains the minor version number of the NI-SCOPE .NET class library.

Implements

[IIVIIdentitySpecificationMinorVersion](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityVendor Property

Gets a string that contains the name of the vendor that supplies the NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Vendor { get; }
```

```
Public ReadOnly Property Vendor As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that contains the name of the vendor that supplies the NI-SCOPE driver.

Implements

IlviComponentIdentityVendor

See Also

Reference

[ScopeDriverIdentity Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentity Methods

The [ScopeDriverIdentity](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object.

	(Inherited from Object .)
 GetGroupCapabilities	Returns a list of names of class capability groups that the IVI specific driver implements.
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetSupportedInstrumentModels	Returns a list of names of instrument models with which the IVI specific driver is compatible.
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityGetGroupCapabilities Method

Returns a list of names of class capability groups that the IVI specific driver implements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string[] GetGroupCapabilities ()
```



```
Public Function GetGroupCapabilities As String()
```

Return Value

Type: [String](#)

An array of [String](#) containing the list of names of class capability groups that the NI-SCOPE driver implements.

Implements

IlviDriverIdentityGetGroupCapabilities

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityGetGroupCapabilities Method

Returns a list of names of class capability groups that the IVI specific driver implements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string[] GetGroupCapabilities()
```

```
Public Function GetGroupCapabilities As String()
```

Return Value

Type: [String](#)

An array of [String](#) containing the list of names of class capability groups that the NI-SCOPE driver implements.

Implements

[IIVIIdentityGetGroupCapabilities](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverIdentityGetSupportedInstrumentModels Method

Returns a list of names of instrument models with which the IVI specific driver is compatible.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string[] GetSupportedInstrumentModels ()
```

```
Public Function GetSupportedInstrumentModels As String()
```

Return Value

Type: [String](#)

An array of [String](#) containing the list of names of instrument models with which the NI-SCOPE is compatible.

Implements

[IIviDriverIdentityGetSupportedInstrumentModels](#)

See Also

Reference

[ScopeDriverIdentity Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLock Class

Provides synchronization locks obtained on the driver session.

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeDriverLock

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)







```
public sealed class ScopeDriverLock : IDisposable,
    IIviDriverLock
```

```
Public NotInheritable Class ScopeDriverLock
    Implements IDisposable, IIviDriverLock
```

The ScopeDriverLock type exposes the following members.

Methods

Name	Description
------	-------------

 Dispose	Releases a driver synchronization lock.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)
 Unlock	Releases a driver synchronization lock.

[Top](#)

Remarks

For more information, refer to [NI High Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLock Methods

The [ScopeDriverLock](#) type exposes the following members.

Methods

Name	Description
 Dispose	Releases a driver synchronization lock.
 Equals	Determines whether the specified object is equal to the current object.

		(Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)
	Unlock	Releases a driver synchronization lock.

[Top](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockDispose Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

[IDisposableDispose](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockDispose Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

[IDisposable.Dispose](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockUnlock Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Unlock()
```

```
Public Sub Unlock
```

Implements

IIviDriverLockUnlock

See Also

Reference

[ScopeDriverLock Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLock Methods

The [ScopeDriverLock](#) type exposes the following members.

Methods

	Name	Description
	Dispose	Releases a driver synchronization lock.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)

 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)
 Unlock	Releases a driver synchronization lock.

[Top](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockDispose Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

[IDisposable.Dispose](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockDispose Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

[IDisposable.Dispose](#)

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverLockUnlock Method

Releases a driver synchronization lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Unlock()
```

```
Public Sub Unlock
```

Implements

IlviDriverLockUnlock

See Also

Reference

[ScopeDriverLock Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation Class

Provides properties that affect the operation of the instrument driver.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeDriverOperation

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax











C#VB

```
public sealed class ScopeDriverOperation : ScopeSubObject,
    IIVIvDriverOperation, IDisposable, ISupportSynchronizationContext
```

```
Public NotInheritable Class ScopeDriverOperation
    Inherits ScopeSubObject
    Implements IIVIvDriverOperation, IDisposable,
    ISupportSynchronizationContext
```






The ScopeDriverOperation type exposes the following members.

Properties

Name	Description
 Cache	Gets or sets whether to cache the value of properties.
 DriverSetup	Gets the driver setup string used to set the initial values for properties that are specific to NI-SCOPE.
 InterchangeCheck	Gets or sets whether to perform interchangeability checking and retrieve interchangeability warnings.
 IOResourceDescriptor	Gets the resource name that the NI-SCOPE uses to identify the physical instrument.
 LogicalName	Gets the logical name that you specified when opening the current session.
 QueryInstrumentStatus	Gets or sets whether NI-SCOPE queries the instrument status after each operation. NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.
 RangeCheck	Gets or sets whether to validate property values and method parameters. If you enable this property, NI-SCOPE validates the parameter values that you pass to NI-SCOPE methods.
 RecordCoercions	Gets or sets whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.
 Simulate	Gets whether or not to simulate NI-SCOPE I/O operations.
 SynchronizeCallbacks	Gets or sets how events and callback delegates are invoked.




[Top](#)

Methods

Name	Description
 Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Events

Name	Description
 Coercion	Occurs when a property is coerced.
 InterchangeCheckWarning	Occurs when an interchange check warning event is raised.
 Warning	Occurs when the driver creates a driver warning.

[Top](#)

Remarks

For more information, refer to [NI High Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference











[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation Properties

The [ScopeDriverOperation](#) type exposes the following members.

Properties

Name	Description
 Cache	Gets or sets whether to cache the value of properties.
 DriverSetup	Gets the driver setup string used to set the initial values for properties that are specific to NI-SCOPE.
 InterchangeCheck	Gets or sets whether to perform interchangeability checking and retrieve interchangeability warnings.
 IOResourceDescriptor	Gets the resource name that the NI-SCOPE uses to identify the physical instrument.
 LogicalName	Gets the logical name that you specified when opening the current session.
 QueryInstrumentStatus	Gets or sets whether NI-SCOPE queries the instrument status after each operation. NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.
 RangeCheck	Gets or sets whether to validate property values and method parameters. If you enable this property, NI-SCOPE validates the parameter values that you pass to NI-SCOPE methods.
 RecordCoercions	Gets or sets whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.
 Simulate	Gets whether or not to simulate NI-SCOPE I/O operations.
 SynchronizeCallbacks	Gets or sets how events and callback delegates are invoked.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCache Property

Gets or sets whether to cache the value of properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Cache { get; set; }
```

```
Public Property Cache As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if caching is enabled; otherwise, . The default value is .

Implements

IlviDriverOperationCache

Remarks

If you set this property to , NI-SCOPE tracks the current NI-SCOPE instrument settings and avoids sending redundant commands to the instrument. NI-SCOPE can either always cache or never cache particular properties, regardless of the setting of this property.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCache Property

Gets or sets whether to cache the value of properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Cache { get; set; }
```

```
Public Property Cache As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if caching is enabled; otherwise, . The default value is .

Implements

IlviDriverOperationCache

Remarks

If you set this property to , NI-SCOPE tracks the current NI-SCOPE instrument settings and avoids sending redundant commands to the instrument. NI-SCOPE can either always cache or never cache particular properties, regardless of the setting of this property.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationDriverSetup Property

Gets the driver setup string used to set the initial values for properties that are specific to NI-SCOPE.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string DriverSetup { get; }
```

```
Public ReadOnly Property DriverSetup As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the driver setup string that you specified in the IVI configuration store.

The default value is an empty string.

Implements

IlviDriverOperationDriverSetup

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationInterchangeCheck Property

Gets or sets whether to perform interchangeability checking and retrieve interchangeability warnings.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool InterchangeCheck { get; set; }
```

```
Public Property InterchangeCheck As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if interchangeability checking is enabled; otherwise .
The default value is .

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationIOResourceDescriptor Property

Gets the resource name that the NI-SCOPE uses to identify the physical instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string IOResourceDescriptor { get; }
```

```
Public ReadOnly Property IOResourceDescriptor As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the resource descriptor that you specified for the physical instrument.

Implements

IlviDriverOperationIOResourceDescriptor

Remarks

If you initialize NI-SCOPE with a logical name, this property contains the resource name that corresponds to the entry in the IVI configuration utility. If you initialize NI-SCOPE with the resource name, this property contains that value.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationLogicalName Property

Gets the logical name that you specified when opening the current session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string LogicalName { get; }
```

```
Public ReadOnly Property LogicalName As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the logical name that you specified when opening the current IVI session.

Implements

IlviDriverOperationLogicalName

Remarks

You may pass a logical name to the NI-SCOPE session during initialization of session. The IVI configuration utility must contain an entry for the logical name. The logical name entry refers to a driver session section in the IVI configuration file. The driver session section specifies a physical instrument and initial options.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationQueryInstrumentStatus Property

Gets or sets whether NI-SCOPE queries the instrument status after each operation. NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool QueryInstrumentStatus { get; set; }
```

```
Public Property QueryInstrumentStatus As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if querying is enabled; otherwise, .

Implements

IlviDriverOperationQueryInstrumentStatus

Remarks

NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.

Querying the instrument status is useful for debugging. After you validate your program, you can set this property to to disable status checking and maximize performance.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationRangeCheck Property

Gets or sets whether to validate property values and method parameters. If you enable this property, NI-SCOPE validates the parameter values that you pass to NI-SCOPE methods.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RangeCheck { get; set; }
```

```
Public Property RangeCheck As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if validating property values and method parameters are enabled; otherwise, .

Implements

IlviDriverOperationRangeCheck

Remarks

Range-checking parameters are very useful for debugging. After you validate your program, you can set this property to to disable range checking and maximize performance.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationRecordCoercions

Property

Gets or sets whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RecordCoercions { get; set; }
```

```
Public Property RecordCoercions As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if the IVI engine keeps a list of the value coercions; otherwise, .

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationSimulate Property

Gets whether or not to simulate NI-SCOPE I/O operations.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Simulate { get; set; }
```

```
Public Property Simulate As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if NI-SCOPE simulates instrument driver I/O operations; if NI-SCOPE communicates directly with the instrument.

Implements

IlviDriverOperationSimulate

Remarks

This property is useful for debugging applications without using hardware. After a session is opened, you cannot change the simulation state.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationSynchronizeCallbacks Property

Gets or sets how events and callback delegates are invoked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool SynchronizeCallbacks { get; set; }
```

```
Public Property SynchronizeCallbacks As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if events and callbacks are invoked through the [Send\(SendOrPostCallback, Object\)](#) or [Post\(SendOrPostCallback, Object\)](#); otherwise, events and callbacks are invoked directly.

The default value is .

Implements

[ISupportSynchronizationContextSynchronizeCallbacks](#)

See Also

Reference

[ScopeDriverOperation Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)










Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation Properties

The [ScopeDriverOperation](#) type exposes the following members.

Properties

Name	Description
 Cache	Gets or sets whether to cache the value of properties.

 DriverSetup	Gets the driver setup string used to set the initial values for properties that are specific to NI-SCOPE.
 InterchangeCheck	Gets or sets whether to perform interchangeability checking and retrieve interchangeability warnings.
 IOResourceDescriptor	Gets the resource name that the NI-SCOPE uses to identify the physical instrument.
 LogicalName	Gets the logical name that you specified when opening the current session.
 QueryInstrumentStatus	Gets or sets whether NI-SCOPE queries the instrument status after each operation. NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.
 RangeCheck	Gets or sets whether to validate property values and method parameters. If you enable this property, NI-SCOPE validates the parameter values that you pass to NI-SCOPE methods.
 RecordCoercions	Gets or sets whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.
 Simulate	Gets whether or not to simulate NI-SCOPE I/O operations.
 SynchronizeCallbacks	Gets or sets how events and callback delegates are invoked.

[Top](#)

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCache Property

Gets or sets whether to cache the value of properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Cache { get; set; }
```

```
Public Property Cache As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if caching is enabled; otherwise, . The default value is .

Implements

IlviDriverOperationCache

Remarks

If you set this property to , NI-SCOPE tracks the current NI-SCOPE instrument settings and avoids sending redundant commands to the instrument. NI-SCOPE can either always cache or never cache particular properties, regardless of the setting of this property.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCache Property

Gets or sets whether to cache the value of properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Cache { get; set; }
```

```
Public Property Cache As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if caching is enabled; otherwise, . The default value is .

Implements

IlviDriverOperationCache

Remarks

If you set this property to , NI-SCOPE tracks the current NI-SCOPE instrument settings and avoids sending redundant commands to the instrument. NI-SCOPE can either always cache or never cache particular properties, regardless of the setting of this property.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationDriverSetup Property

Gets the driver setup string used to set the initial values for properties that are specific to NI-SCOPE.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string DriverSetup { get; }
```

```
Public ReadOnly Property DriverSetup As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the driver setup string that you specified in the IVI configuration store.

The default value is an empty string.

Implements

IlviDriverOperationDriverSetup

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationInterchangeCheck Property

Gets or sets whether to perform interchangeability checking and retrieve interchangeability warnings.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool InterchangeCheck { get; set; }
```

```
Public Property InterchangeCheck As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if interchangeability checking is enabled; otherwise .
The default value is .

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationIOResourceDescriptor Property

Gets the resource name that the NI-SCOPE uses to identify the physical instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string IOResourceDescriptor { get; }
```

```
Public ReadOnly Property IOResourceDescriptor As String
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the resource descriptor that you specified for the physical instrument.

Implements

[IlviDriverOperationIOResourceDescriptor](#)

Remarks

If you initialize NI-SCOPE with a logical name, this property contains the resource name that corresponds to the entry in the IVI configuration utility. If you initialize NI-SCOPE with the resource name, this property contains that value.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationLogicalName Property

Gets the logical name that you specified when opening the current session.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in

[NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public string LogicalName { get; }
```

```
Public ReadOnly Property LogicalName As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the logical name that you specified when opening the current IVI session.

Implements

IlviDriverOperationLogicalName

Remarks

You may pass a logical name to the NI-SCOPE session during initialization of session. The IVI configuration utility must contain an entry for the logical name. The logical name entry refers to a driver session section in the IVI configuration file. The driver session section specifies a physical instrument and initial options.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationQueryInstrumentStatus Property

Gets or sets whether NI-SCOPE queries the instrument status after each operation. NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool QueryInstrumentStatus { get; set; }
```

```
Public Property QueryInstrumentStatus As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if querying is enabled; otherwise, .

Implements

IlviDriverOperationQueryInstrumentStatus

Remarks

NI-SCOPE can choose to ignore status checking for particular properties, regardless of the setting of this property.

Querying the instrument status is useful for debugging. After you validate your program, you can set this property to to disable status checking and maximize performance.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationRangeCheck Property

Gets or sets whether to validate property values and method parameters. If you enable this property, NI-SCOPE validates the parameter values that you pass to NI-SCOPE methods.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RangeCheck { get; set; }
```

```
Public Property RangeCheck As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if validating property values and method parameters are enabled; otherwise, .

Implements

[IIVIDriverOperationRangeCheck](#)

Remarks

Range-checking parameters are very useful for debugging. After you validate your program, you can set this property to to disable range checking and maximize performance.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationRecordCoercions Property

Gets or sets whether the IVI engine keeps a list of the value coercions it makes for integer and real type properties.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RecordCoercions { get; set; }
```

```
Public Property RecordCoercions As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if the IVI engine keeps a list of the value coercions; otherwise, .

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationSimulate Property

Gets whether or not to simulate NI-SCOPE I/O operations.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Simulate { get; set; }
```

```
Public Property Simulate As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if NI-SCOPE simulates instrument driver I/O operations; if NI-SCOPE communicates directly with the instrument.

Implements

IlviDriverOperationSimulate

Remarks

This property is useful for debugging applications without using hardware. After a session is opened, you cannot change the simulation state.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeDriverOperationSynchronizeCallbacks Property](#)

Gets or sets how events and callback delegates are invoked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool SynchronizeCallbacks { get; set; }
```

```
Public Property SynchronizeCallbacks As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if events and callbacks are invoked through the [Send\(SendOrPostCallback, Object\)](#) or [Post\(SendOrPostCallback, Object\)](#); otherwise, events and callbacks are invoked directly.

The default value is .

Implements

ISupportSynchronizationContextSynchronizeCallbacks

See Also

Reference

[ScopeDriverOperation Class](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation Methods

The [ScopeDriverOperation](#) type exposes the following members.

Methods

Name	Description
 Dispose	Closes the specified session and deallocates the reserved resources, if not already disposed.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationDispose Method

Closes the specified session and deallocates the reserved resources, if not already disposed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

Public Sub Dispose

Implements

IDisposable.Dispose

Remarks

You can call this method safely more than once, even if the session is already closed. A call to this method disposes the SafeHandle class used to hold the instrument handle. If the call to this method fails due to some reason, such as the session being closed by some external means, you will not be notified about the failure. To help you identify failures in the ReleaseHandle method of the SafeHandle class, managed debugging assistant (MDA) is activated.

For details, refer to <http://msdn.microsoft.com/en-us/library/85eak4a0.aspx>.

The call to this method fails when you externally close a session by:

- Initializing a session with a resource name for which the session is already open, within the same process. This causes the instrument handle held by the existing session to become invalid.
- Getting the instrument handle out using [GetInstrumentHandle](#) and closing this handle directly.

See Also

Reference

ScopeDriverOperation Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation.Dispose Method

Closes the specified session and deallocates the reserved resources, if not already disposed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Dispose()
```

```
Public Sub Dispose
```

Implements

[IDisposable.Dispose](#)

Remarks

You can call this method safely more than once, even if the session is already closed. A call to this method disposes the SafeHandle class used to hold the instrument handle. If the call to this method fails due to some reason, such as the session being closed by some external means, you will not be notified about the failure. To help you identify failures in the ReleaseHandle method of the SafeHandle class, managed debugging assistant (MDA) is activated.

For details, refer to <http://msdn.microsoft.com/en-us/library/85eak4a0.aspx>.

The call to this method fails when you externally close a session by:

- Initializing a session with a resource name for which the session is already open, within the same process. This causes the instrument handle held by the existing session to become invalid.
- Getting the instrument handle out using [GetInstrumentHandle](#) and closing this handle directly.

See Also

Reference

[ScopeDriverOperation Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperation Events

The [ScopeDriverOperation](#) type exposes the following members.

Events

Name	Description
 Coercion	Occurs when a property is coerced.
 InterchangeCheckWarning	Occurs when an interchange check warning event is raised.
 Warning	Occurs when the driver creates a driver warning.

[Top](#)

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCoercion Event

Occurs when a property is coerced.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public event EventHandler<ScopeCoercionEventArgs> Coercion
```

```
Public Event Coercion As EventHandler(Of ScopeCoercionEventArgs)
```

Value

Type: [SystemEventHandlerScopeCoercionEventArgs](#)

Remarks

This event is triggered only if [RecordCoercions](#) is set to true.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationCoercion Event

Occurs when a property is coerced.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public event EventHandler<ScopeCoercionEventArgs> Coercion
```

```
Public Event Coercion As EventHandler(Of ScopeCoercionEventArgs)
```

Value

Type: [SystemEventHandlerScopeCoercionEventArgs](#)

Remarks

This event is triggered only if [RecordCoercions](#) is set to true.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationInterchangeCheckWarning Event

Occurs when an interchange check warning event is raised.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public event EventHandler<ScopeInterchangeCheckWarningEventArgs>
InterchangeCheckWarning
```

```
Public Event InterchangeCheckWarning As EventHandler(Of
ScopeInterchangeCheckWarningEventArgs)
```

Value

Type: [SystemEventHandlerScopeInterchangeCheckWarningEventArgs](#)

Remarks

This event is triggered only if [InterchangeCheck](#) is set to .

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverOperationWarning Event

Occurs when the driver creates a driver warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public event EventHandler<ScopeWarningEventArgs> Warning
```

```
Public Event Warning As EventHandler(Of ScopeWarningEventArgs)
```

Value

Type: [SystemEventHandlerScopeWarningEventArgs](#)

Remarks

You will receive event notifications on registration as listeners.

See Also

Reference

[ScopeDriverOperation Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtility Class

Contains methods that provide a basic set of utility operations.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeDriverUtility

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeDriverUtility : ScopeSubObject,
    IIVIUtility, IDisposable
```

```
Public NotInheritable Class ScopeDriverUtility
    Inherits ScopeSubObject
    Implements IIVIUtility, IDisposable
```


The ScopeDriverUtility type exposes the following members.

Properties

	Name	Description
	<u>FpgaBitfilePath</u>	Gets the absolute file path to the bitfile loaded on the FPGA.

Top

Methods

	Name	Description
	<u>Disable</u>	Places the instrument in a quiescent state as quickly as possible.

Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
ExportAttributeConfigurationBuffer	Exports the attribute configuration of the session to a Byte array buffer.
ExportAttributeConfigurationFile	Exports the attribute configuration of the session to the specified file.
GetChannelName	Returns the channel name given a Int32 one-based index.
GetChannelNameFromString	Returns a comma-separated channel-name list given a String index list.
GetHashCode	Serves as the default hash function. (Inherited from Object .)
GetType	Gets the Type of the current instance. (Inherited from Object .)
ImportAttributeConfigurationBuffer	Imports the attribute configuration to the session from the Byte array configuration.
ImportAttributeConfigurationFile	Imports an attribute configuration to the session from the specified file.
Lock	Attempts to acquire a synchronization lock on this instance of the driver.
Lock(PrecisionTimeSpan)	Attempts to acquire a synchronization lock on this instance of the driver and specifies the maximum amount of time to wait to acquire the lock.
QueryDriverRevision	Returns the revision numbers of the NI-SCOPE instrument driver.
Reset	Resets all properties to their default value and stops export of all external signals and events.
ResetDevice	Performs a hard reset on the instrument.
SelfTest	Performs a self-test on the NI-SCOPE instrument and returns the test result.
ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtility Properties

The [ScopeDriverUtility](#) type exposes the following members.

Properties

	Name	Description
	FpgaBitfilePath	Gets the absolute file path to the bitfile loaded on the FPGA.

[Top](#)

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityFpgaBitfilePath Property

Gets the absolute file path to the bitfile loaded on the FPGA.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FpgaBitfilePath { get; }
```

```
Public ReadOnly Property FpgaBitfilePath As String
    Get
```

Property Value

Type: [String](#)

The absolute file path to the bitfile loaded on the FPGA.

Exceptions

Exception	Condition
ObjectDisposedException	FpgaBitfilePath was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	FpgaBitfilePath was accessed in a session to an unsupported instrument.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

For more information see [NI-SCOPE Instrument Driver FPGA Extensions](#) in the NI High-Speed Digitizers Help.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityFpgaBitfilePath Property

Gets the absolute file path to the bitfile loaded on the FPGA.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FpgaBitfilePath { get; }
```

```
Public ReadOnly Property FpgaBitfilePath As String
    Get
```

Property Value

Type: [String](#)

The absolute file path to the bitfile loaded on the FPGA.

Exceptions

Exception	Condition
ObjectDisposedException	FpgaBitfilePath was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	FpgaBitfilePath was accessed in a session to an unsupported instrument.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

For more information see [NI-SCOPE Instrument Driver FPGA Extensions](#) in the NI High-Speed Digitizers Help.

See Also

Reference

[ScopeDriverUtility Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtility Properties

The [ScopeDriverUtility](#) type exposes the following members.

Properties

	Name	Description
	FpgaBitfilePath	Gets the absolute file path to the bitfile loaded on the FPGA.

[Top](#)

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityFpgaBitfilePath Property

Gets the absolute file path to the bitfile loaded on the FPGA.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FpgaBitfilePath { get; }
```

```
Public ReadOnly Property FpgaBitfilePath As String
    Get
```

Property Value

Type: [String](#)

The absolute file path to the bitfile loaded on the FPGA.

Exceptions

Exception	Condition
ObjectDisposedException	FpgaBitfilePath was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	FpgaBitfilePath was accessed in a session to an unsupported instrument.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

For more information see [NI-SCOPE Instrument Driver FPGA Extensions](#) in the NI High-Speed Digitizers Help.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityFpgaBitfilePath Property

Gets the absolute file path to the bitfile loaded on the FPGA.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FpgaBitfilePath { get; }
```

```
Public ReadOnly Property FpgaBitfilePath As String
    Get
```

Property Value

Type: [String](#)

The absolute file path to the bitfile loaded on the FPGA.

Exceptions

Exception	Condition
ObjectDisposedException	FpgaBitfilePath was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
OperationNotSupportedException	FpgaBitfilePath was accessed in a session to an unsupported instrument.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

For more information see [NI-SCOPE Instrument Driver FPGA Extensions](#) in the NI High-Speed Digitizers Help.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtility Methods

The [ScopeDriverUtility](#) type exposes the following members.

Methods

Name	Description
------	-------------

Disable	Places the instrument in a quiescent state as quickly as possible.
Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
ExportAttributeConfigurationBuffer	Exports the attribute configuration of the session to a Byte array buffer.
ExportAttributeConfigurationFile	Exports the attribute configuration of the session to the specified file.
GetChannelName	Returns the channel name given a Int32 one-based index.
GetChannelNameFromString	Returns a comma-separated channel-name list given a String index list.
GetHashCode	Serves as the default hash function. (Inherited from Object .)
GetType	Gets the Type of the current instance. (Inherited from Object .)
ImportAttributeConfigurationBuffer	Imports the attribute configuration to the session from the Byte array configuration.
ImportAttributeConfigurationFile	Imports an attribute configuration to the session from the specified file.
Lock	Attempts to acquire a synchronization lock on this instance of the driver.
Lock(PrecisionTimeSpan)	Attempts to acquire a synchronization lock on this instance of the driver and specifies the maximum amount of time to wait to acquire the lock.
QueryDriverRevision	Returns the revision numbers of the NI-SCOPE instrument driver.
Reset	Resets all properties to their default value and stops export of all external signals and events.
ResetDevice	Performs a hard reset on the instrument.
SelfTest	Performs a self-test on the NI-SCOPE instrument and returns the test result.
ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityDisable Method

Places the instrument in a quiescent state as quickly as possible.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Disable()
```

```
Public Sub Disable
```

Remarks

In a quiescent state, an instrument has no or minimal effect on the external system to which it is connected.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityDisable Method

Places the instrument in a quiescent state as quickly as possible.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Disable()
```

```
Public Sub Disable
```

Remarks

In a quiescent state, an instrument has no or minimal effect on the external system to which it is connected.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityExportAttributeConfigurationBuffer Method

Exports the attribute configuration of the session to a [Byte](#) array buffer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public byte[] ExportAttributeConfigurationBuffer()
```

```
Public Function ExportAttributeConfigurationBuffer As Byte()
```

Return Value

Type: [Byte](#)

Returns a [Byte](#) array populated with the exported attribute configuration.

Exceptions

Exception	Condition
ObjectDisposedException	The <code>ExportAttributeConfigurationBuffer</code> method was called after the associated NIScope object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can export and import session attribute configurations only between devices with identical model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

This method verifies that the attributes you have configured for the session are valid. If the configuration is invalid, NI-SCOPE returns an error.

When importing and exporting configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the "resourceName" input to the [NIScope](#) constructor. For example, if your entry for "resourceName" is "PXI1Slot1,PXI1Slot2" for the exporting session and "PXI2Slot2,PXI2Slot3" for the importing session:

- The configuration exported from PXI1Slot1 is imported into PXI2Slot2.
- The configuration exported from PXI1Slot2 is imported into PXI2Slot3.

Note

NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityExportAttributeConfigurationFile Method

Exports the attribute configuration of the session to the specified file.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ExportAttributeConfigurationFile(  
    string filePath  
)
```

```
Public Sub ExportAttributeConfigurationFile (  
    filePath As String  
)
```

Parameters

filePath

Type: [SystemString](#)

The absolute path to the file to contain the exported attribute configuration. If you specify an empty or relative path, this method returns an error. The default file extension is `.niscopeconfig`.

Exceptions

Exception	Condition
ObjectDisposedException	The <code>ExportAttributeConfigurationFile(String)</code> method was called after the associated NIScope object was disposed.
<code>IviDriverException</code>	The underlying NI-SCOPE driver returned an error.

Remarks

You can export and import session attribute configurations only between devices with identical model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

This method verifies that the attributes you have configured for the session are valid. If the configuration is invalid, NI-SCOPE returns an error.

When importing and exporting configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the "resourceName" input to the [NIScope](#) constructor. For example, if your entry for "resourceName" is "PXI1Slot1,PXI1Slot2" for the exporting session and "PXI2Slot2,PXI2Slot3" for the importing session:

- The configuration exported from PXI1Slot1 is imported into PXI2Slot2.
- The configuration exported from PXI1Slot2 is imported into PXI2Slot3.

Note

NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityGetChannelName Method

Returns the channel name given a [Int32](#) one-based index.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string GetChannelName(
    int index
)
```

```
Public Function GetChannelName (
    index As Integer
) As String
```

Parameters

index

Type: [SystemInt32](#)

One-based index for the desired channel in the session. Valid values are from 1 to the number of channels in the session

Return Value

Type: [String](#)

Returns a [String](#) populated with the channel name.

Exceptions

Exception	Condition
ObjectDisposedException	The GetChannelName(Int32) method was called after the associated NIScope or ScopeDriverUtility object was disposed.
OutOfRangeException	The indicated index is out of range (less than zero or greater than the number of channels in the session)

Remarks

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityGetChannelNameFromString Method

Returns a comma-separated channel-name list given a [String](#) index list.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public string GetChannelNameFromString(
    string index
)
```

```
Public Function GetChannelNameFromString (
    index As String
) As String
```

Parameters

index

Type: [SystemString](#)

Index list for the channels in the session. Valid values are from 0 to the number of channels in the session minus 1. The input string can be in one of the following formats: "0,2,3,1" !-zdocsholder-!> Comma-separated list "0-3" !-zdocsholder-!> Range using a hyphen "0:3" !-zdocsholder-!> Range using a colon - The above formats can be combined: "1, 1:2, 0-3" - Out-of-

order indices are supported: "2,3,0" - Repeated indices are supported: "1, 2, 2, 3, 0" - White-space characters (space, tab, newline, vertical tab, feed, carriage return) are allowed between characters: " 1,2", "1 , 2 ", " 1 ,2 ", "1, 2", "1 , 2" are valid strings - Ranges can be incrementing (0-3) or decrementing (3-0)

Return Value

Type: [String](#)

Returns a [String](#) populated with the corresponding name for each index in the same order the indices appear in the input string.

Exceptions

Exception	Condition
ObjectDisposedException	The <code>GetChannelNameFromString(String)</code> method was called after the associated NIScope or ScopeDriverUtility object was disposed.
OutOfRangeException	If one of the indices in the list is greater than or equal to the number of channels in the session.
SelectorNameException	If the input list does not have a valid format or contains a negative index.
IviCDriverException	If the input list is empty.

Remarks

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityImportAttributeConfigurationBuffer Method

Imports the attribute configuration to the session from the [Byte](#) array configuration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ImportAttributeConfigurationBuffer(
    byte[] configuration
)
```

```
Public Sub ImportAttributeConfigurationBuffer (
    configuration As Byte()
)
```

Parameters

configuration

Type: [SystemByte](#)

The byte array that contains the attribute configuration to import.

Exceptions

Exception	Condition
ObjectDisposedException	The ImportAttributeConfigurationBuffer(Byte) method was called after the associated NIScope object was disposed.
NullReferenceException	The buffer or its underlying data have been disposed
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can export and import session attribute configurations only between devices with identical model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

Note

You cannot call this method while the session is in a running state, such as while acquiring a signal.

When importing and exporting configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the "resourceName" input to the [NIScope](#) constructor. For example, if your entry for "resourceName" is "PXI1Slot1,PXI1Slot2" for the exporting session and "PXI2Slot2,PXI2Slot3" for the importing session:

- The configuration exported from PXI1Slot1 is imported into PXI2Slot2.
- The configuration exported from PXI1Slot2 is imported into PXI2Slot3.

Note

NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityImportAttributeConfigurationFile Method

Imports an attribute configuration to the session from the specified file.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ImportAttributeConfigurationFile(  
    string filePath  
)
```

```
Public Sub ImportAttributeConfigurationFile (
    filePath As String
)
```

Parameters

filePath

Type: [SystemString](#)

The absolute path to the file containing the attribute configuration to import. If you specify an empty or relative path, this method returns an error. The default file extension is .niscopeconfig.

Exceptions

Exception	Condition
ObjectDisposedException	The ImportAttributeConfigurationFile(String) method was called after the associated NIScope object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can export and import session attribute configurations only between devices with identical model numbers, channel counts, and onboard memory sizes and between NI-SCOPE sessions with the same number of initialized channels.

Note

You cannot call this method while the session is in a running state, such as while acquiring a signal.

When importing and exporting configurations between NI-SCOPE sessions that were both initialized with multiple instruments, the configurations of the exporting instruments are mapped to the importing instruments in the order you specify in the "resourceName" input to the [NIScope](#) constructor. For example, if your entry for "resourceName" is "PXI1Slot1,PXI1Slot2" for the exporting session and "PXI2Slot2,PXI2Slot3" for the importing session:

- The configuration exported from PXI1Slot1 is imported into PXI2Slot2.
- The configuration exported from PXI1Slot2 is imported into PXI2Slot3.

 Note

NI-SCOPE will return an error if the total number of channels initialized for the exporting session is not equal to the total number of channels initialized for the importing session.

See Also

Reference



[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityLock Method

Overload List

Name	Description
 Lock	Attempts to acquire a synchronization lock on this instance of the driver.
 Lock(PrecisionTimeSpan)	Attempts to acquire a synchronization lock on this instance of the driver and specifies the maximum amount of time to wait to acquire the lock.

[Top](#)

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityLock Method

Attempts to acquire a synchronization lock on this instance of the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverLock Lock()
```

```
Public Function Lock As ScopeDriverLock
```

Return Value

Type: [ScopeDriverLock](#)

A reference to the acquired lock.

See Also

Reference

[ScopeDriverUtility Class](#)

[Lock Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityLock Method

Attempts to acquire a synchronization lock on this instance of the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverLock Lock()
```

```
Public Function Lock As ScopeDriverLock
```

Return Value

Type: [ScopeDriverLock](#)

A reference to the acquired lock.

See Also

Reference

[ScopeDriverUtility Class](#)

[Lock Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityLock Method (PrecisionTimeSpan)

Attempts to acquire a synchronization lock on this instance of the driver and specifies the maximum amount of time to wait to acquire the lock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDriverLock Lock(
    PrecisionTimeSpan maxTime
)
```

```
Public Function Lock (
    maxTime As PrecisionTimeSpan
) As ScopeDriverLock
```

Parameters

maxTime

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The maximum amount of time to wait to acquire the lock.

Return Value

Type: [ScopeDriverLock](#)

A reference to the acquired lock.

See Also

Reference

[ScopeDriverUtility Class](#)

[Lock Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityQueryDriverRevision Method

Returns the revision numbers of the NI-SCOPE instrument driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public ScopeRevisionQueryResult QueryDriverRevision()
```

```
Public Function QueryDriverRevision As ScopeRevisionQueryResult
```

Return Value

Type: [ScopeRevisionQueryResult](#)

An object of type [ScopeRevisionQueryResult](#).

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityReset Method

Resets all properties to their default value and stops export of all external signals and events.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Reset()
```

```
Public Sub Reset
```

Implements

[IIVIUtilityReset](#)

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilityResetDevice Method

Performs a hard reset on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ResetDevice()
```

```
Public Sub ResetDevice
```

Remarks

Resets the instrument to a known state. The method disables power generation, resets session properties to their default values, properties clears errors such as overtemperature and unexpected loss of auxiliary power, commits the session properties, and leaves the session in the Uncommitted state. This method also performs a hard reset on the instrument and driver software. This method has the same functionality as using reset in Measurement and Automation Explorer. This will also open the output relay on instruments that have an output relay.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeDriverUtilitySelfTest Method

Performs a self-test on the NI-SCOPE instrument and returns the test result.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSelfTestResult SelfTest()
```

```
Public Function SelfTest As ScopeSelfTestResult
```

Return Value

Type: [ScopeSelfTestResult](#)

An object of type SelfTestResult.

Remarks

This method performs a simple series of tests verifying that the NI-SCOPE instrument is powered on and responding.

Note

This method calls [Reset](#), which resets the software state.

See Also

Reference

[ScopeDriverUtility Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTrigger Class

Represents the sub-object that contains methods and properties used to configure common properties for analog edge triggering.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeEdgeTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeEdgeTrigger : ScopeSubObject,
    IIVI_ScopeTriggerEdge
```

```
Public NotInheritable Class ScopeEdgeTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTriggerEdge
```





The ScopeEdgeTrigger type exposes the following members.


Properties

	Name	Description
	<u>Slope</u>	Specifies whether a rising or a falling edge triggers the digitizer.

Top

Methods

	Name	Description
	<u>Configure</u>	Configures common properties for analog edge triggering.
	<u>Equals</u>	Determines whether the specified object is equal to the current object. (Inherited from <u>Object</u> .)
	<u>GetHashCode</u>	Serves as the default hash function. (Inherited from <u>Object</u> .)
	<u>GetType</u>	Gets the <u>Type</u> of the current instance. (Inherited from <u>Object</u> .)

 ToString	Returns a string that represents the current object. (Inherited from Object .)
--	---

[Top](#)

Remarks

For more information, refer to the Trigger related topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTrigger Properties

The [ScopeEdgeTrigger](#) type exposes the following members.

Properties

	Name	Description
	Slope	Specifies whether a rising or a falling edge triggers the digitizer.

[Top](#)

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerSlope Property

Specifies whether a rising or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerSlope Slope { get; set; }
```

```
Public Property Slope As ScopeTriggerSlope  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerSlope](#)

Specifies the [ScopeTriggerSlope](#).

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerSlope Property

Specifies whether a rising or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerSlope Slope { get; set; }
```

```
Public Property Slope As ScopeTriggerSlope
    Get
    Set
```

Property Value

Type: [ScopeTriggerSlope](#)

Specifies the [ScopeTriggerSlope](#).

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTrigger Properties

The [ScopeEdgeTrigger](#) type exposes the following members.

Properties

Name	Description
------	-------------

[Slope](#)

Specifies whether a rising or a falling edge triggers the digitizer.

[Top](#)[See Also](#)[Reference](#)[ScopeEdgeTrigger Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerSlope Property

Specifies whether a rising or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

[Syntax](#)[C#VB](#)

```
public ScopeTriggerSlope Slope { get; set; }
```

```
Public Property Slope As ScopeTriggerSlope  
    Get  
    Set
```

[Property Value](#)Type: [ScopeTriggerSlope](#)Specifies the [ScopeTriggerSlope](#).[See Also](#)[Reference](#)[ScopeEdgeTrigger Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerSlope Property

Specifies whether a rising or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerSlope Slope { get; set; }
```

```
Public Property Slope As ScopeTriggerSlope  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerSlope](#)

Specifies the [ScopeTriggerSlope](#).

See Also

Reference

ScopeEdgeTrigger Class






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTrigger Methods

The [ScopeEdgeTrigger](#) type exposes the following members.

Methods

Name	Description
 Configure	Configures common properties for analog edge triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerConfigure Method

Configures common properties for analog edge triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public void Configure(
    ScopeTriggerSource triggerSource,
    double triggerLevel,
    ScopeTriggerSlope triggerSlope,
```

```

    ScopeTriggerCoupling triggerCoupling,
    PrecisionTimeSpan triggerHoldoff,
    PrecisionTimeSpan triggerDelay
)

```

```

Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    triggerLevel As Double,
    triggerSlope As ScopeTriggerSlope,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
 The trigger source. Refer to [ScopeTriggerSource](#) for defined values.

triggerLevel

Type: [SystemDouble](#)
 The voltage threshold for the trigger. Refer to [Level](#) for more information.

triggerSlope

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSlope](#)
 A value indicating whether you want a rising edge or a falling edge to trigger the digitizer. Refer to [ScopeTriggerSlope](#) for defined values.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)
 Applies coupling and filtering options to the trigger signal. Refer to [ScopeTriggerCoupling](#) for defined values.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)
 The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to [Holdoff](#) for more information.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)
 The length of time the digitizer waits after receiving the trigger to start acquiring data. Refer to [Delay](#) for more information.

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEdgeTriggerConfigure Method

Configures common properties for analog edge triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    double triggerLevel,  
    ScopeTriggerSlope triggerSlope,  
    ScopeTriggerCoupling triggerCoupling,  
    PrecisionTimeSpan triggerHoldoff,  
    PrecisionTimeSpan triggerDelay  
)
```

```
Public Sub Configure (  
    triggerSource As ScopeTriggerSource,  
    triggerLevel As Double,  
    triggerSlope As ScopeTriggerSlope,  
    triggerCoupling As ScopeTriggerCoupling,  
    triggerHoldoff As PrecisionTimeSpan,  
    triggerDelay As PrecisionTimeSpan  
)
```


Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source. Refer to [ScopeTriggerSource](#) for defined values.

triggerLevel

Type: [SystemDouble](#)

The voltage threshold for the trigger. Refer to [Level](#) for more information.

triggerSlope

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSlope](#)

A value indicating whether you want a rising edge or a falling edge to trigger the digitizer. Refer to [ScopeTriggerSlope](#) for defined values.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

Applies coupling and filtering options to the trigger signal. Refer to [ScopeTriggerCoupling](#) for defined values.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to [Holdoff](#) for more information.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time the digitizer waits after receiving the trigger to start acquiring data. Refer to [Delay](#) for more information.

See Also

Reference

[ScopeEdgeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Class

Represents the output terminal for [EndOfAcquisitionEventOutputTerminal](#).

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax















C#VB









```
public sealed class ScopeEndOfAcquisitionEventOutputTerminal
```

```
Public NotInheritable Class ScopeEndOfAcquisitionEventOutputTerminal
```

The ScopeEndOfAcquisitionEventOutputTerminal type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.

 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.





[Top](#)

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeEndOfAcquisitionEventOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfAcquisitionEventOutputTerminal)	Determines whether the current instance of <code>ScopeEndOfAcquisitionEventOutputTerminal</code> and the specified <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeEndOfAcquisitionEventOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeEndOfAcquisitionEventOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether two <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object instances are equal.
	(String to <code>ScopeEndOfAcquisitionEventOutputTerminal</code>)	Converts a specified String to an equivalent <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object.
	(<code>ScopeEndOfAcquisitionEventOutputTerminal</code> to String)	Converts a <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object to an equivalent String .
	Inequality	Checks whether two <code>ScopeEndOfAcquisitionEventOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference























[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsEndOfAcquisitionEventOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Properties

The [ScopeEndOfAcquisitionEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)
 A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi5

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)
 A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalClockOut

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi0 Pro

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi1 Pro

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi2

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPxiStar

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As
    ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi0 Pr

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi1 Pr

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi2 Pr

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi4 Pr

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi5 Pr

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi6 Pr

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As  
ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi7 Pr

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```


Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSL_7".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)















[NationalInstruments.ModularInstruments.NIScope Namespace](#)









Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Properties

The [ScopeEndOfAcquisitionEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.

 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
    ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeEndOfAcquisitionEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)
 A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As
ScopeEndOfAcquisitionEventOutputTerminal
    Get
```


Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
ScopeEndOfAcquisitionEventOutputTerminal  
Get
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

A [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Methods

The [ScopeEndOfAcquisitionEventOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfAcquisitionEventOutputTerminal)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified ScopeEndOfAcquisitionEventOutputTerminal object are equal.



 FromString	Creates a ScopeEndOfAcquisitionEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeEndOfAcquisitionEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeEndOfAcquisitionEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)[See Also](#)[Reference](#)[ScopeEndOfAcquisitionEventOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals M

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfAcquisitionEventOutputTerminal)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified ScopeEndOfAcquisitionEventOutputTerminal object are equal.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals

Determines whether the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals M

Determines whether the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```


Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals M

Determines whether the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified [ScopeEndOfAcquisitionEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeEndOfAcquisitionEventOutputTerminal source  
)
```

```
Public Function Equals (
    source As ScopeEndOfAcquisitionEventOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
 Specifies the [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfAcquisitionEventOutputTerminal)	Determines whether the current instance of ScopeEndOfAcquisitionEventOutputTerminal and the specified

[ScopeEndOfAcquisitionEventOutputTerminal](#) object are equal.

[Top](#)

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals

Determines whether the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquals Method
(ScopeEndOfAcquisitionEventOutputTerminal)

Determines whether the current instance of

[ScopeEndOfAcquisitionEventOutputTerminal](#) and the specified

[ScopeEndOfAcquisitionEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(
    ScopeEndOfAcquisitionEventOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeEndOfAcquisitionEventOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
 Specifies the [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare to the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalFromString Method

Creates a [ScopeEndOfAcquisitionEventOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfAcquisitionEventOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeEndOfAcquisitionEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeEndOfAcquisitionEventOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

Returns an object of type [ScopeEndOfAcquisitionEventOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal.GetHashCode Method

Gets the hash code for the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalToString Method

Converts the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeEndOfAcquisitionEventOutputTerminal](#).

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Operators and Type Conversions

The [ScopeEndOfAcquisitionEventOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeEndOfAcquisitionEventOutputTerminal object instances are equal.
	(String to ScopeEndOfAcquisitionEventOutputTerminal)	Converts a specified String to an equivalent ScopeEndOfAcquisitionEventOutputTerminal object.

 	(ScopeEndOfAcquisitionEventOutputTerminal to String)	Converts a ScopeEndOfAcquisitionEventOutputTerminal object to an equivalent String .
 	Inequality	Checks whether two ScopeEndOfAcquisitionEventOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquality

Checks whether two [ScopeEndOfAcquisitionEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeEndOfAcquisitionEventOutputTerminal source1,
    ScopeEndOfAcquisitionEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeEndOfAcquisitionEventOutputTerminal,
    source2 As ScopeEndOfAcquisitionEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
Specifies the first [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
Specifies the second [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalEquality Operator

Checks whether two [ScopeEndOfAcquisitionEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeEndOfAcquisitionEventOutputTerminal source1,
```

```
ScopeEndOfAcquisitionEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeEndOfAcquisitionEventOutputTerminal,
    source2 As ScopeEndOfAcquisitionEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
 Specifies the first [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
 Specifies the second [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Conversion Operators

Overload List

Name	Description
------	-------------

 	(String to ScopeEndOfAcquisitionEventOutputTerminal)	Converts a specified String to an equivalent ScopeEndOfAcquisitionEventOutputTerminal object.
 	(ScopeEndOfAcquisitionEventOutputTerminal to String)	Converts a ScopeEndOfAcquisitionEventOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Conversion (String to ScopeEndOfAcquisitionEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeEndOfAcquisitionEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeEndOfAcquisitionEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeEndOfAcquisitionEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeEndOfAcquisitionEventOutputTerminal](#) object.

Return Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

Returns the [ScopeEndOfAcquisitionEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Conversion (String to ScopeEndOfAcquisitionEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeEndOfAcquisitionEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeEndOfAcquisitionEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeEndOfAcquisitionEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeEndOfAcquisitionEventOutputTerminal](#) object.

Return Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

Returns the [ScopeEndOfAcquisitionEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminal Conversion
(ScopeEndOfAcquisitionEventOutputTerminal to String)

Converts a [ScopeEndOfAcquisitionEventOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeEndOfAcquisitionEventOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeEndOfAcquisitionEventOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
Specifies the [ScopeEndOfAcquisitionEventOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeEndOfAcquisitionEventOutputTerminal](#) object.

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)
[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfAcquisitionEventOutputTerminalInequality Operator

Checks whether two [ScopeEndOfAcquisitionEventOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeEndOfAcquisitionEventOutputTerminal source1,
    ScopeEndOfAcquisitionEventOutputTerminal source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeEndOfAcquisitionEventOutputTerminal,
    source2 As ScopeEndOfAcquisitionEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
Specifies the first [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfAcquisitionEventOutputTerminal](#)
Specifies the second [ScopeEndOfAcquisitionEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeEndOfAcquisitionEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Class

Represents the end of record event output terminal for [EndOfRecordEventOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB




















```
public sealed class ScopeEndOfRecordEventOutputTerminal
```

```
Public NotInheritable Class ScopeEndOfRecordEventOutputTerminal
```

The `ScopeEndOfRecordEventOutputTerminal` type exposes the following members.



Properties






	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.

 Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
 Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
 Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
 Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
 Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
 ClockOut	Gets the output terminal for CLK OUT.
 NoSource	Gets the output terminal for no source.
 Pfi0	Gets the output terminal for PFI 0.
 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)









Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeEndOfRecordEventOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfRecordEventOutputTerminal)	Determines whether the current instance of <code>ScopeEndOfRecordEventOutputTerminal</code> and the specified <code>ScopeEndOfRecordEventOutputTerminal</code> object are equal.

 FromString 	Creates a ScopeEndOfRecordEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeEndOfRecordEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeEndOfRecordEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two ScopeEndOfRecordEventOutputTerminal object instances are equal.
 	(String to ScopeEndOfRecordEventOutputTerminal)	Converts a specified String to an equivalent ScopeEndOfRecordEventOutputTerminal object.
 	(ScopeEndOfRecordEventOutputTerminal to String)	Converts a ScopeEndOfRecordEventOutputTerminal object to an equivalent String .
 	Inequality	Checks whether two ScopeEndOfRecordEventOutputTerminal object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference













[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignalsScopeExportedSignalsEndOfRecordEventOutputTerminal](#)











Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Properties

The [ScopeEndOfRecordEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.

 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi0 Pr

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi0 Pr

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi1 Pr

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi2 Pr

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi3 Pr

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi4 Pr

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi5 Pr

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi6 Pr

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi7 Pr

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtSi0

Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSl_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)









[NationalInstruments.ModularInstruments.NIScope Namespace](#)















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Properties

The [ScopeEndOfRecordEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.

 ClockOut	Gets the output terminal for CLK OUT.
 NoSource	Gets the output terminal for no source.
 Pfi0	Gets the output terminal for PFI 0.
 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi0 Pr

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeEndOfRecordEventOutputTerminal
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeEndOfRecordEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

A [ScopeEndOfRecordEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Methods

The [ScopeEndOfRecordEventOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfRecordEventOutputTerminal)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified ScopeEndOfRecordEventOutputTerminal object are equal.
 FromString 	Creates a ScopeEndOfRecordEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeEndOfRecordEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeEndOfRecordEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Meth

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfRecordEventOutputTerminal)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified ScopeEndOfRecordEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Meth

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Meth

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Meth

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified [ScopeEndOfRecordEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeEndOfRecordEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeEndOfRecordEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the [ScopeEndOfRecordEventOutputTerminal](#) object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeEndOfRecordEventOutputTerminal)	Determines whether the current instance of ScopeEndOfRecordEventOutputTerminal and the specified ScopeEndOfRecordEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Meth

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquals Method
(ScopeEndOfRecordEventOutputTerminal)

Determines whether the current instance of [ScopeEndOfRecordEventOutputTerminal](#) and the specified [ScopeEndOfRecordEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeEndOfRecordEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeEndOfRecordEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the [ScopeEndOfRecordEventOutputTerminal](#) object to compare to the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeEndOfRecordEventOutputTerminalFromString Method](#)

Creates a [ScopeEndOfRecordEventOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeEndOfRecordEventOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeEndOfRecordEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeEndOfRecordEventOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

Returns an object of type [ScopeEndOfRecordEventOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalGetHashCode Method

Gets the hash code for the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalToString Method

Converts the current instance of [ScopeEndOfRecordEventOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeEndOfRecordEventOutputTerminal](#).

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Operators and Type Conversions

The [ScopeEndOfRecordEventOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeEndOfRecordEventOutputTerminal object instances are equal.
	(String to ScopeEndOfRecordEventOutputTerminal)	Converts a specified String to an equivalent ScopeEndOfRecordEventOutputTerminal object.
	(ScopeEndOfRecordEventOutputTerminal to String)	Converts a ScopeEndOfRecordEventOutputTerminal object to an equivalent String .



Inequality

Checks whether two [ScopeEndOfRecordEventOutputTerminal](#) object instances are not equal.

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquality Operator

Checks whether two [ScopeEndOfRecordEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeEndOfRecordEventOutputTerminal source1,
    ScopeEndOfRecordEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeEndOfRecordEventOutputTerminal,
    source2 As ScopeEndOfRecordEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the first [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the second [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalEquality Operator

Checks whether two [ScopeEndOfRecordEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeEndOfRecordEventOutputTerminal source1,
```

```
ScopeEndOfRecordEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeEndOfRecordEventOutputTerminal,
    source2 As ScopeEndOfRecordEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
 Specifies the first [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
 Specifies the second [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Conversion Operators

Overload List

Name	Description
------	-------------

 	(String to ScopeEndOfRecordEventOutputTerminal)	Converts a specified String to an equivalent ScopeEndOfRecordEventOutputTerminal object.
 	(ScopeEndOfRecordEventOutputTerminal to String)	Converts a ScopeEndOfRecordEventOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Conversion (String to ScopeEndOfRecordEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeEndOfRecordEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeEndOfRecordEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeEndOfRecordEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeEndOfRecordEventOutputTerminal](#) object.

Return Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

Returns the [ScopeEndOfRecordEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Conversion (String to ScopeEndOfRecordEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeEndOfRecordEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeEndOfRecordEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeEndOfRecordEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeEndOfRecordEventOutputTerminal](#) object.

Return Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

Returns the [ScopeEndOfRecordEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminal Conversion
(ScopeEndOfRecordEventOutputTerminal to String)

Converts a [ScopeEndOfRecordEventOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeEndOfRecordEventOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeEndOfRecordEventOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the [ScopeEndOfRecordEventOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeEndOfRecordEventOutputTerminal](#) object.

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeEndOfRecordEventOutputTerminalInequality Operator

Checks whether two [ScopeEndOfRecordEventOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeEndOfRecordEventOutputTerminal source1,  
    ScopeEndOfRecordEventOutputTerminal source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeEndOfRecordEventOutputTerminal,  
    source2 As ScopeEndOfRecordEventOutputTerminal  
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the first [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeEndOfRecordEventOutputTerminal](#)
Specifies the second [ScopeEndOfRecordEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeEndOfRecordEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult Structure

Represents the the result of an error query.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public struct ScopeErrorQueryResult
```

```
Public Structure ScopeErrorQueryResult
```

The ScopeErrorQueryResult type exposes the following members.






Properties

	Name	Description
	Code	Gets the error code returned by the driver.
	Message	Gets the readable error string.

Top



Methods

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeErrorQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeErrorQueryResult)	Determines whether the current instance of ScopeErrorQueryResult and the ScopeErrorQueryResult object that you specify are equal.
 GetHashCode	Returns the hash code for the current instance of ScopeErrorQueryResult. (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

Operators

	Name	Description
 Equality	Equality	Checks whether the two instances of ScopeErrorQueryResult are equal.
 Inequality	Inequality	Checks whether the two instances of ScopeErrorQueryResult are unequal.

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult Properties

The [ScopeErrorQueryResult](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the error code returned by the driver.
	Message	Gets the readable error string.

[Top](#)

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultCode Property

Gets the error code returned by the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) representing the error code.

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultCode Property

Gets the error code returned by the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) representing the error code.

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultMessage Property

Gets the readable error string.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Message { get; }
```

```
Public ReadOnly Property Message As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the error string.

See Also

Reference

[ScopeErrorQueryResult Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult Properties

The [ScopeErrorQueryResult](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the error code returned by the driver.
	Message	Gets the readable error string.

[Top](#)

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultCode Property

Gets the error code returned by the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) representing the error code.

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultCode Property

Gets the error code returned by the driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) representing the error code.

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultMessage Property

Gets the readable error string.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Message { get; }
```

```
Public ReadOnly Property Message As String
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the error string.

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult Methods

The [ScopeErrorQueryResult](#) type exposes the following members.

Methods

Name	Description
------	-------------



 Equals(Object)	Determines whether the current instance of ScopeErrorQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeErrorQueryResult)	Determines whether the current instance of ScopeErrorQueryResult and the ScopeErrorQueryResult object that you specify are equal.
 GetHashCode	Returns the hash code for the current instance of ScopeErrorQueryResult . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)[See Also](#)[Reference](#)[ScopeErrorQueryResult Structure](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeErrorQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeErrorQueryResult)	Determines whether the current instance of ScopeErrorQueryResult and the ScopeErrorQueryResult object that you specify are equal.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (Object)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (Object)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (ScopeErrorQueryResult)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the [ScopeErrorQueryResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeErrorQueryResult result  
)
```

```
Public Function Equals (
    result As ScopeErrorQueryResult
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
The [ScopeErrorQueryResult](#) object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeErrorQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeErrorQueryResult)	Determines whether the current instance of ScopeErrorQueryResult and the ScopeErrorQueryResult object that you specify are equal.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (Object)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (Object)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquals Method (ScopeErrorQueryResult)

Determines whether the current instance of [ScopeErrorQueryResult](#) and the [ScopeErrorQueryResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeErrorQueryResult result  
)
```



```
Public Function Equals (
    result As ScopeErrorQueryResult
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
The [ScopeErrorQueryResult](#) object to be compared to the current instance of [ScopeErrorQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult.GetHashCode Method

Returns the hash code for the current instance of [ScopeErrorQueryResult](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) representing the hash value generated for the current instance of [ScopeErrorQueryResult](#).

See Also

Reference

[ScopeErrorQueryResult Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResult Operators

The [ScopeErrorQueryResult](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether the two instances of ScopeErrorQueryResult are equal.
	Inequality	Checks whether the two instances of ScopeErrorQueryResult are unequal.

[Top](#)

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquality Operator

Checks whether the two instances of [ScopeErrorQueryResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeErrorQueryResult result1,
    ScopeErrorQueryResult result2
)
```

```
Public Shared Operator = (
    result1 As ScopeErrorQueryResult,
    result2 As ScopeErrorQueryResult
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultEquality Operator

Checks whether the two instances of [ScopeErrorQueryResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeErrorQueryResult result1,
    ScopeErrorQueryResult result2
)
```

```
Public Shared Operator = (
    result1 As ScopeErrorQueryResult,
    result2 As ScopeErrorQueryResult
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeErrorQueryResultInequality Operator

Checks whether the two instances of [ScopeErrorQueryResult](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeErrorQueryResult result1,
    ScopeErrorQueryResult result2
)
```

```
Public Shared Operator <> (
    result1 As ScopeErrorQueryResult,
    result2 As ScopeErrorQueryResult
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeErrorQueryResult](#)
A [ScopeErrorQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeErrorQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignals Class

Represents the ScopeExportedSignals sub-objects that are used to configure the driver to generate a signal that other instruments can detect when configuring for digital triggering or sharing clocks.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax











[C#](#)[VB](#)










```
public sealed class ScopeExportedSignals : ScopeSubObject
```

```
Public NotInheritable Class ScopeExportedSignals
    Inherits ScopeSubObject
```

The ScopeExportedSignals type exposes the following members.





Properties

Name	Description
 AdvanceTriggerOutputTerminal	Gets or sets the destination terminal to which advanceTrigger is exported.
 AdvanceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Advance Trigger. When the Advance Trigger is received, the digitizer begins acquiring samples for the Nth record.
 EndOfAcquisitionEventOutputTerminal	Gets or sets the destination for the end of acquisition event.
 EndOfAcquisitionEventTerminalName	Specifies the terminal name for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records.
 EndOfRecordEventOutputTerminal	Gets or sets the destination for the end of record event.
 EndOfRecordEventTerminalName	Specifies the terminal name for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record.
 FiveVoltOutputTerminal	Gets or sets the destination for the exported 5 Volt signal.
 ReadyForAdvanceEventOutputTerminal	Gets or sets the destination for the Ready for Advance trigger event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForAdvanceEventTerminalName	Specifies the terminal name for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForReferenceEventOutputTerminal	Gets or sets the destination for the Ready for Reference trigger event. When this event is asserted, the digitizer is ready to receive a reference trigger.

 ReadyForReferenceEventTerminalName	Specifies the terminal name for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.
 ReadyForStartEventOutputTerminal	Gets or sets the destination for the Ready for Start trigger event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReadyForStartEventTerminalName	Specifies the terminal name for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReferenceClockOutputTerminal	Gets or sets the output terminal to which the reference clock is exported.
 ReferenceTriggerOutputTerminal	Gets or sets the destination to which the reference (Stop) trigger is exported.
 ReferenceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Reference Trigger.
 SampleClockOutputTerminal	Gets or sets the sample clock for the specified terminal.
 StartTriggerOutputTerminal	Gets or sets the destination to which the start trigger is exported.
 StartTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Start Trigger. When the Start Trigger is received, the digitizer begins acquiring samples.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the `niScope_ExportSignal` topic in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources




[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)















Copyright © National Instruments Corporation. All Rights Reserved.



ScopeExportedSignals Properties

The [ScopeExportedSignals](#) type exposes the following members.

Properties

Name	Description
 AdvanceTriggerOutputTerminal	Gets or sets the destination terminal to which <code>advanceTrigger</code> is exported.
 AdvanceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Advance Trigger. When the Advance Trigger is received, the digitizer begins acquiring samples for the Nth record.
 EndOfAcquisitionEventOutputTerminal	Gets or sets the destination for the end of acquisition event.

 EndOfAcquisitionEventTerminalName	Specifies the terminal name for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records.
 EndOfRecordEventOutputTerminal	Gets or sets the destination for the end of record event.
 EndOfRecordEventTerminalName	Specifies the terminal name for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record.
 FiveVoltOutputTerminal	Gets or sets the destination for the exported 5 Volt signal.
 ReadyForAdvanceEventOutputTerminal	Gets or sets the destination for the Ready for Advance trigger event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForAdvanceEventTerminalName	Specifies the terminal name for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForReferenceEventOutputTerminal	Gets or sets the destination for the Ready for Reference trigger event. When this event is asserted, the digitizer is ready to receive a reference trigger.
 ReadyForReferenceEventTerminalName	Specifies the terminal name for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.
 ReadyForStartEventOutputTerminal	Gets or sets the destination for the Ready for Start trigger event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReadyForStartEventTerminalName	Specifies the terminal name for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReferenceClockOutputTerminal	Gets or sets the output terminal to which the reference clock is exported.
 ReferenceTriggerOutputTerminal	Gets or sets the destination to which the reference (Stop) trigger is exported.
 ReferenceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Reference Trigger.
 SampleClockOutputTerminal	Gets or sets the sample clock for the specified terminal.

 StartTriggerOutputTerminal	Gets or sets the destination to which the start trigger is exported.
 StartTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Start Trigger. When the Start Trigger is received, the digitizer begins acquiring samples.

[Top](#)

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerOutputTerminal

Gets or sets the destination terminal to which advanceTrigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerExportedOutputTerminal AdvanceTriggerOutputTerminal
{ get; set; }
```

```
Public Property AdvanceTriggerOutputTerminal As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

An object of type [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerOutputTerminal

Gets or sets the destination terminal to which advanceTrigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerExportedOutputTerminal AdvanceTriggerOutputTerminal
{ get; set; }
```

```
Public Property AdvanceTriggerOutputTerminal As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

An object of type [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerTerminalName

Specifies the terminal name for the output trigger signal of the Advance Trigger. When the Advance Trigger is received, the digitizer begins acquiring samples for the Nth record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string AdvanceTriggerTerminalName { get; }
```

```
Public ReadOnly Property AdvanceTriggerTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Advance Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfAcquisitionEventOutputTerminal

Gets or sets the destination for the end of acquisition event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEndOfAcquisitionEventOutputTerminal  
EndOfAcquisitionEventOutputTerminal { get; set; }
```

```
Public Property EndOfAcquisitionEventOutputTerminal As  
ScopeEndOfAcquisitionEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

An object of type [ScopeEndOfAcquisitionEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfAcquisitionEventTerminalName

Specifies the terminal name for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string EndOfAcquisitionEventTerminalName { get; }
```

```
Public ReadOnly Property EndOfAcquisitionEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the End of Acquisition Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfRecordEventOutputTerminal

Gets or sets the destination for the end of record event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEndOfRecordEventOutputTerminal EndOfRecordEventOutputTerminal
{ get; set; }
```

```
Public Property EndOfRecordEventOutputTerminal As
ScopeEndOfRecordEventOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

An object of type [ScopeEndOfRecordEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfRecordEventTerminalName

Specifies the terminal name for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string EndOfRecordEventTerminalName { get; }
```

```
Public ReadOnly Property EndOfRecordEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the End of Record Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsFiveVoltOutputTerminal Property

Gets or sets the destination for the exported 5 Volt signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFiveVoltsOutOutputTerminal FiveVoltOutputTerminal { get; set; }
```

```
Public Property FiveVoltOutputTerminal As ScopeFiveVoltsOutOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

An object of type [ScopeFiveVoltsOutOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForAdvanceEventOutput

Gets or sets the destination for the Ready for Advance trigger event. When this event is asserted, the digitizer is ready to receive an advance trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForAdvanceEventOutputTerminal  
ReadyForAdvanceEventOutputTerminal { get; set; }
```

```
Public Property ReadyForAdvanceEventOutputTerminal As  
ScopeReadyForAdvanceEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

An object of type [ScopeReadyForAdvanceEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForAdvanceEventTerminalName

Specifies the terminal name for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForAdvanceEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForAdvanceEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Advance Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForReferenceEventOutputTerminal

Gets or sets the destination for the Ready for Reference trigger event. When this event is asserted, the digitizer is ready to receive a reference trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForReferenceEventOutputTerminal  
ReadyForReferenceEventOutputTerminal { get; set; }
```

```
Public Property ReadyForReferenceEventOutputTerminal As  
ScopeReadyForReferenceEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

An object of type [ScopeReadyForReferenceEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForReferenceEventTerminalName

Specifies the terminal name for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForReferenceEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForReferenceEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Reference Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForStartEventOutputTerminal

Gets or sets the destination for the Ready for Start trigger event. When this event is asserted, the digitizer is ready to receive a start trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForStartEventOutputTerminal ReadyForStartEventOutputTerminal  
{ get; set; }
```

```
Public Property ReadyForStartEventOutputTerminal As  
ScopeReadyForStartEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

An object of type [ScopeReadyForStartEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForStartEventTerminalName

Specifies the terminal name for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForStartEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForStartEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Start Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceClockOutputTerminalName

Gets or sets the output terminal to which the reference clock is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeOutputClockSource ReferenceClockOutputTerminal { get; set; }
```

```
Public Property ReferenceClockOutputTerminal As ScopeOutputClockSource
    Get
    Set
```

Property Value

Type: [ScopeOutputClockSource](#)

An object of type [ScopeOutputClockSource](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceTriggerOutputTerminal

Gets or sets the destination to which the reference (Stop) trigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTriggerExportedOutputTerminal
ReferenceTriggerOutputTerminal { get; set; }
```

```
Public Property ReferenceTriggerOutputTerminal As
ScopeReferenceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

An object of type [ScopeReferenceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceTriggerTerminalName

Specifies the terminal name for the output trigger signal of the Reference Trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReferenceTriggerTerminalName { get; }
```

```
Public ReadOnly Property ReferenceTriggerTerminalName As String
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Reference Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsSampleClockOutputTerminal P

Gets or sets the sample clock for the specified terminal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleClockExportedOutputTerminal SampleClockOutputTerminal { get;
set; }
```

```
Public Property SampleClockOutputTerminal As
ScopeSampleClockExportedOutputTerminal
```

Get
Set

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

An object of type [ScopeSampleClockExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsStartTriggerOutputTerminal Property

Gets or sets the destination to which the start trigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeStartTriggerExportedOutputTerminal StartTriggerOutputTerminal
{ get; set; }
```

```
Public Property StartTriggerOutputTerminal As
ScopeStartTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

An object of type [ScopeStartTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsStartTriggerTerminalName Property

Specifies the terminal name for the output trigger signal of the Start Trigger. When the Start Trigger is received, the digitizer begins acquiring samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string StartTriggerTerminalName { get; }
```

```
Public ReadOnly Property StartTriggerTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Start Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources








[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)













Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignals Properties

The [ScopeExportedSignals](#) type exposes the following members.

Properties

Name	Description
 AdvanceTriggerOutputTerminal	Gets or sets the destination terminal to which advanceTrigger is exported.
 AdvanceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Advance Trigger. When the Advance Trigger is received, the digitizer begins acquiring samples for the Nth record.
 EndOfAcquisitionEventOutputTerminal	Gets or sets the destination for the end of acquisition event.
 EndOfAcquisitionEventTerminalName	Specifies the terminal name for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records.
 EndOfRecordEventOutputTerminal	Gets or sets the destination for the end of record event.
 EndOfRecordEventTerminalName	Specifies the terminal name for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record.
 FiveVoltOutputTerminal	Gets or sets the destination for the exported 5 Volt signal.

 ReadyForAdvanceEventOutputTerminal	Gets or sets the destination for the Ready for Advance trigger event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForAdvanceEventTerminalName	Specifies the terminal name for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.
 ReadyForReferenceEventOutputTerminal	Gets or sets the destination for the Ready for Reference trigger event. When this event is asserted, the digitizer is ready to receive a reference trigger.
 ReadyForReferenceEventTerminalName	Specifies the terminal name for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.
 ReadyForStartEventOutputTerminal	Gets or sets the destination for the Ready for Start trigger event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReadyForStartEventTerminalName	Specifies the terminal name for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.
 ReferenceClockOutputTerminal	Gets or sets the output terminal to which the reference clock is exported.
 ReferenceTriggerOutputTerminal	Gets or sets the destination to which the reference (Stop) trigger is exported.
 ReferenceTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Reference Trigger.
 SampleClockOutputTerminal	Gets or sets the sample clock for the specified terminal.
 StartTriggerOutputTerminal	Gets or sets the destination to which the start trigger is exported.
 StartTriggerTerminalName	Specifies the terminal name for the output trigger signal of the Start Trigger. When the Start Trigger is received, the digitizer begins acquiring samples.

[Top](#)

[See Also](#)

[Reference](#)

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerOutputTerminal

Gets or sets the destination terminal to which advanceTrigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerExportedOutputTerminal AdvanceTriggerOutputTerminal
{ get; set; }
```

```
Public Property AdvanceTriggerOutputTerminal As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

An object of type [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerOutputTerminal Property

Gets or sets the destination terminal to which advanceTrigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTriggerExportedOutputTerminal AdvanceTriggerOutputTerminal
{ get; set; }
```

```
Public Property AdvanceTriggerOutputTerminal As
ScopeAdvanceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeAdvanceTriggerExportedOutputTerminal](#)

An object of type [ScopeAdvanceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsAdvanceTriggerTerminalName Property

Specifies the terminal name for the output trigger signal of the Advance Trigger. When the Advance Trigger is received, the digitizer begins acquiring samples for the Nth record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string AdvanceTriggerTerminalName { get; }
```

```
Public ReadOnly Property AdvanceTriggerTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Advance Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeExportedSignalsEndOfAcquisitionEventOutputTerminal](#) Property

Gets or sets the destination for the end of acquisition event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEndOfAcquisitionEventOutputTerminal  
EndOfAcquisitionEventOutputTerminal { get; set; }
```

```
Public Property EndOfAcquisitionEventOutputTerminal As  
ScopeEndOfAcquisitionEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeEndOfAcquisitionEventOutputTerminal](#)

An object of type [ScopeEndOfAcquisitionEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfAcquisitionEventTerminalName Property

Specifies the terminal name for the End of Acquisition Event. When this event is asserted, the digitizer has completed sampling for all records.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string EndOfAcquisitionEventTerminalName { get; }
```

```
Public ReadOnly Property EndOfAcquisitionEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the End of Acquisition Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfRecordEventOutputTerminal Property

Gets or sets the destination for the end of record event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEndOfRecordEventOutputTerminal EndOfRecordEventOutputTerminal
{ get; set; }
```

```
Public Property EndOfRecordEventOutputTerminal As
ScopeEndOfRecordEventOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeEndOfRecordEventOutputTerminal](#)

An object of type [ScopeEndOfRecordEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsEndOfRecordEventTerminalName Property

Specifies the terminal name for the End of Record Event. When this event is asserted, the digitizer has completed sampling for the current record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string EndOfRecordEventTerminalName { get; }
```

```
Public ReadOnly Property EndOfRecordEventTerminalName As String
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the End of Record Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsFiveVoltOutputTerminal Property

Gets or sets the destination for the exported 5 Volt signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFiveVoltsOutOutputTerminal FiveVoltOutputTerminal { get; set; }
```

```
Public Property FiveVoltOutputTerminal As ScopeFiveVoltsOutOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

An object of type [ScopeFiveVoltsOutOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForAdvanceEventOutputTerminal Property

Gets or sets the destination for the Ready for Advance trigger event. When this event is asserted, the digitizer is ready to receive an advance trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForAdvanceEventOutputTerminal  
ReadyForAdvanceEventOutputTerminal { get; set; }
```

```
Public Property ReadyForAdvanceEventOutputTerminal As  
ScopeReadyForAdvanceEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

An object of type [ScopeReadyForAdvanceEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForAdvanceEventTerminalName Property

Specifies the terminal name for the Ready for Advance Event. When this event is asserted, the digitizer is ready to receive an advance trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForAdvanceEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForAdvanceEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Advance Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForReferenceEventOutputTerminal Property

Gets or sets the destination for the Ready for Reference trigger event. When this event is asserted, the digitizer is ready to receive a reference trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForReferenceEventOutputTerminal  
ReadyForReferenceEventOutputTerminal { get; set; }
```

```
Public Property ReadyForReferenceEventOutputTerminal As  
ScopeReadyForReferenceEventOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

An object of type [ScopeReadyForReferenceEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForReferenceEventTerminalName Property

Specifies the terminal name for the Ready for Reference Event. When this event is asserted, the digitizer is ready to receive a reference trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForReferenceEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForReferenceEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Reference Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForStartEventOutputTerminal Property

Gets or sets the destination for the Ready for Start trigger event. When this event is asserted, the digitizer is ready to receive a start trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReadyForStartEventOutputTerminal ReadyForStartEventOutputTerminal
{ get; set; }
```

```
Public Property ReadyForStartEventOutputTerminal As
ScopeReadyForStartEventOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

An object of type [ScopeReadyForStartEventOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReadyForStartEventTerminalName Property

Specifies the terminal name for the Ready for Start Event. When this event is asserted, the digitizer is ready to receive a start trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReadyForStartEventTerminalName { get; }
```

```
Public ReadOnly Property ReadyForStartEventTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Ready for Start Event Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceClockOutputTerminal Property

Gets or sets the output terminal to which the reference clock is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeOutputClockSource ReferenceClockOutputTerminal { get; set; }
```

```
Public Property ReferenceClockOutputTerminal As ScopeOutputClockSource  
    Get  
    Set
```

Property Value

Type: [ScopeOutputClockSource](#)

An object of type [ScopeOutputClockSource](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceTriggerOutputTerminal Property

Gets or sets the destination to which the reference (Stop) trigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTriggerExportedOutputTerminal
ReferenceTriggerOutputTerminal { get; set; }
```

```
Public Property ReferenceTriggerOutputTerminal As
ScopeReferenceTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

An object of type [ScopeReferenceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsReferenceTriggerTerminalName Property

Specifies the terminal name for the output trigger signal of the Reference Trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string ReferenceTriggerTerminalName { get; }
```

```
Public ReadOnly Property ReferenceTriggerTerminalName As String  
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Reference Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsSampleClockOutputTerminal Property

Gets or sets the sample clock for the specified terminal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleClockExportedOutputTerminal SampleClockOutputTerminal { get; set; }
```

```
Public Property SampleClockOutputTerminal As  
ScopeSampleClockExportedOutputTerminal  
    Get  
    Set
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

An object of type [ScopeSampleClockExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsStartTriggerOutputTerminal Property

Gets or sets the destination to which the start trigger is exported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeStartTriggerExportedOutputTerminal StartTriggerOutputTerminal
{ get; set; }
```

```
Public Property StartTriggerOutputTerminal As
ScopeStartTriggerExportedOutputTerminal
    Get
    Set
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

An object of type [ScopeStartTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignalsStartTriggerTerminalName Property

Specifies the terminal name for the output trigger signal of the Start Trigger. When the Start Trigger is received, the digitizer begins acquiring samples.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string StartTriggerTerminalName { get; }
```

```
Public ReadOnly Property StartTriggerTerminalName As String
    Get
```

Property Value

Type: [String](#)

A string containing the fully qualified name for the Start Trigger Terminal.

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExportedSignals Methods

The [ScopeExportedSignals](#) type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeExportedSignals Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibration Class

Represents methods used to externally calibrate digitizer.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeExternalCalibration

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public sealed class ScopeExternalCalibration : ScopeSubObject
```

```
Public NotInheritable Class ScopeExternalCalibration
    Inherits ScopeSubObject
```





The ScopeExternalCalibration type exposes the following members.

Properties

	Name	Description
	ExternalCalibrationDate	Gets the last external calibration date.
	ExternalCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of external calibration.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Calibration topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources



[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibration Properties

The [ScopeExternalCalibration](#) type exposes the following members.

Properties

Name	Description
 ExternalCalibrationDate	Gets the last external calibration date.
 ExternalCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of external calibration.

[Top](#)

See Also

Reference

[ScopeExternalCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationDate Pro

Gets the last external calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime ExternalCalibrationDate { get; }
```

```
Public ReadOnly Property ExternalCalibrationDate As DateTime
    Get
```

Property Value

Type: [DateTime](#)

Last external calibration date.

See Also

Reference

[ScopeExternalCalibration Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationDate Pro

Gets the last external calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime ExternalCalibrationDate { get; }
```

```
Public ReadOnly Property ExternalCalibrationDate As DateTime  
    Get
```

Property Value

Type: [DateTime](#)

Last external calibration date.

See Also

Reference

[ScopeExternalCalibration Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationTemperature

Gets the onboard temperature of the digitizer at the time of external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ExternalCalibrationTemperature { get; }
```

```
Public ReadOnly Property ExternalCalibrationTemperature As Double  
    Get
```

Property Value

Type: [Double](#)

Onboard temperature of the digitizer at the time of external calibration.

See Also

Reference

[ScopeExternalCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources



[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibration Properties

The [ScopeExternalCalibration](#) type exposes the following members.

Properties

Name	Description
 ExternalCalibrationDate	Gets the last external calibration date.
 ExternalCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of external calibration.

[Top](#)

See Also

Reference

[ScopeExternalCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationDate Pro

Gets the last external calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime ExternalCalibrationDate { get; }
```

```
Public ReadOnly Property ExternalCalibrationDate As DateTime
    Get
```


Property Value

Type: [DateTime](#)

Last external calibration date.

See Also

Reference

[ScopeExternalCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationDate Property

Gets the last external calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public DateTime ExternalCalibrationDate { get; }
```

```
Public ReadOnly Property ExternalCalibrationDate As DateTime  
    Get
```

Property Value

Type: [DateTime](#)

Last external calibration date.

See Also

Reference

[ScopeExternalCalibration Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibrationExternalCalibrationTemperature Property

Gets the onboard temperature of the digitizer at the time of external calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ExternalCalibrationTemperature { get; }
```

```
Public ReadOnly Property ExternalCalibrationTemperature As Double  
    Get
```

Property Value

Type: [Double](#)

Onboard temperature of the digitizer at the time of external calibration.

See Also

Reference

[ScopeExternalCalibration Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeExternalCalibration Methods

The [ScopeExternalCalibration](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeExternalCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFetchRelativeTo Enumeration

Specifies the position to start fetching within one record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeFetchRelativeTo
```

```
Public Enumeration ScopeFetchRelativeTo
```

Members

Member name	Value	Description
Start	482	Fetches data starting at the first point sampled by the digitizer. For devices that support continuous acquisition, you must fetch relative to Now or relative to Start before the trigger occurs. The number of points available when fetching relative to start may be extremely large because there may be an infinite number of pretrigger samples if the trigger never occurs. Furthermore, the pretrigger points are sampled into a circular buffer, so the data at the start may be overwritten. If the data is no longer available for fetching, the fetch method throws an exception.
Trigger	483	Starts fetching at the first posttrigger sample, which is useful for fetching a small number of points near the trigger.
Pretrigger	477	Fetches data relative to the first pretrigger point requested.
Now	481	Starts fetching data at the last sample acquired. Therefore, you must use a negative offset when fetching relative to now. This option is useful for configuring an infinite acquisition (one that never triggers) and occasionally fetching the most recent data.
ReadPointer	388	Specifies the read pointer value. When you initiate a new acquisition, the read pointer is set to 0. At this point, fetching relative to the read pointer is equivalent to fetching relative to start. After every fetch, the read pointer is incremented to be the sample after the last sample retrieved. Therefore, you can repeatedly fetch relative to the read pointer for a continuous acquisition program.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Class

Represents the output terminal for [FiveVoltOutputTerminal](#).

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax











C#VB













```
public sealed class ScopeFiveVoltsOutOutputTerminal
```

```
Public NotInheritable Class ScopeFiveVoltsOutOutputTerminal
```

The ScopeFiveVoltsOutOutputTerminal type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.

	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of <code>ScopeFiveVoltsOutOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeFiveVoltsOutOutputTerminal)	Determines whether the current instance of <code>ScopeFiveVoltsOutOutputTerminal</code> and the specified <code>ScopeFiveVoltsOutOutputTerminal</code> object are equal.
 	FromString	Creates a <code>ScopeFiveVoltsOutOutputTerminal</code> object from the specified String .
	GetHashCode	Gets the hash code for the current instance of <code>ScopeFiveVoltsOutOutputTerminal</code> . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of <code>ScopeFiveVoltsOutOutputTerminal</code> to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether two <code>ScopeFiveVoltsOutOutputTerminal</code> object instances are equal.
	(String to ScopeFiveVoltsOutOutputTerminal)	Converts a specified <code>String</code> to an equivalent <code>ScopeFiveVoltsOutOutputTerminal</code> object.
	(ScopeFiveVoltsOutOutputTerminal to String)	Converts a <code>ScopeFiveVoltsOutOutputTerminal</code> object to an equivalent <code>String</code> .
	Inequality	Checks whether two <code>ScopeFiveVoltsOutOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)























[ScopeExportedSignalsFiveVoltOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Properties

The [ScopeFiveVoltsOutOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	<u>Aux0Pfi0</u>	Gets the output terminal for AUX I/O connector 0 PFI 0.
	<u>Aux0Pfi1</u>	Gets the output terminal for AUX I/O connector 0 PFI 1.
	<u>Aux0Pfi2</u>	Gets the output terminal for AUX I/O connector 0 PFI 2.
	<u>Aux0Pfi3</u>	Gets the output terminal for AUX I/O connector 0 PFI 3.
	<u>Aux0Pfi4</u>	Gets the output terminal for AUX I/O connector 0 PFI 4.
	<u>Aux0Pfi5</u>	Gets the output terminal for AUX I/O connector 0 PFI 5.
	<u>Aux0Pfi6</u>	Gets the output terminal for AUX I/O connector 0 PFI 6.
	<u>Aux0Pfi7</u>	Gets the output terminal for AUX I/O connector 0 PFI 7.
	<u>ClockOut</u>	Gets the output terminal for CLK OUT.
	<u>NoSource</u>	Gets the output terminal for no source.
	<u>Pfi0</u>	Gets the output terminal for PFI 0.
	<u>Pfi1</u>	Gets the output terminal for PFI 1.
	<u>Pfi2</u>	Gets the output terminal for PFI 2.
	<u>PxiStar</u>	Gets the output terminal for PXI STAR.
	<u>Rtsi0</u>	Gets the output terminal for RTSI 0.
	<u>Rtsi1</u>	Gets the output terminal for RTSI 1.
	<u>Rtsi2</u>	Gets the output terminal for RTSI 2.
	<u>Rtsi3</u>	Gets the output terminal for RTSI 3.
	<u>Rtsi4</u>	Gets the output terminal for RTSI 4.
	<u>Rtsi5</u>	Gets the output terminal for RTSI 5.
	<u>Rtsi6</u>	Gets the output terminal for RTSI 6.
	<u>Rtsi7</u>	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal ClockOut { get; }
```



```
Public Shared ReadOnly Property ClockOut As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtSi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal RtSi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RT5I_4".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RT5I_5".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)




















[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Properties

The [ScopeFiveVoltsOutOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.

 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeFiveVoltsOutOutputTerminal
    Get
```


Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: ScopeFiveVoltsOutOutputTerminal

A ScopeFiveVoltsOutOutputTerminal representing the String "VAL_AUX_0_PFI_5".

See Also

Reference

ScopeFiveVoltsOutOutputTerminal Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeFiveVoltsOutOutputTerminal
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeFiveVoltsOutOutputTerminal  
    Get
```

Property Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

A [ScopeFiveVoltsOutOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Methods

The [ScopeFiveVoltsOutOutputTerminal](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeFiveVoltsOutOutputTerminal)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified ScopeFiveVoltsOutOutputTerminal object are equal.
 	FromString	Creates a ScopeFiveVoltsOutOutputTerminal object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeFiveVoltsOutOutputTerminal . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeFiveVoltsOutOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)[See Also](#)[Reference](#)[ScopeFiveVoltsOutOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method

Overload List

	Name	Description
=	Equals(Object)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
=	Equals(ScopeFiveVoltsOutOutputTerminal)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified ScopeFiveVoltsOutOutputTerminal object are equal.

[Top](#)[See Also](#)[Reference](#)[ScopeFiveVoltsOutOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeFiveVoltsOutOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

Equals Overload

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method (Object)

Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified object are equal.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: SystemObject

Specifies the object to compare to the current instance of ScopeFiveVoltsOutOutputTerminal.

Return Value

Type: Boolean

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method (ScopeFiveVoltsOutOutputTerminal)

Determines whether the current instance of [ScopeFiveVoltsOutOutputTerminal](#) and the specified [ScopeFiveVoltsOutOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeFiveVoltsOutOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeFiveVoltsOutOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the [ScopeFiveVoltsOutOutputTerminal](#) object to compare to the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeFiveVoltsOutOutputTerminal)	Determines whether the current instance of ScopeFiveVoltsOutOutputTerminal and the specified ScopeFiveVoltsOutOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeFiveVoltsOutOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeFiveVoltsOutOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquals Method

(ScopeFiveVoltsOutOutputTerminal)

Determines whether the current instance of [ScopeFiveVoltsOutOutputTerminal](#) and the specified [ScopeFiveVoltsOutOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public bool Equals(  
    ScopeFiveVoltsOutOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeFiveVoltsOutOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the [ScopeFiveVoltsOutOutputTerminal](#) object to compare to the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalFromString Method

Creates a [ScopeFiveVoltsOutOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeFiveVoltsOutOutputTerminal FromString(  
    string source  
)
```



```
Public Shared Function FromString (
    source As String
) As ScopeFiveVoltsOutOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeFiveVoltsOutOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

Returns an object of type [ScopeFiveVoltsOutOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalGetHashCode Method

Gets the hash code for the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalToString Method

Converts the current instance of [ScopeFiveVoltsOutOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeFiveVoltsOutOutputTerminal](#).

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Operators and Type Conversions

The [ScopeFiveVoltsOutOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeFiveVoltsOutOutputTerminal object instances are equal.
	(String to ScopeFiveVoltsOutOutputTerminal)	Converts a specified String to an equivalent ScopeFiveVoltsOutOutputTerminal object.
	(ScopeFiveVoltsOutOutputTerminal to String)	Converts a ScopeFiveVoltsOutOutputTerminal object to an equivalent String .
	Inequality	Checks whether two ScopeFiveVoltsOutOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquality Operator

Checks whether two [ScopeFiveVoltsOutOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeFiveVoltsOutOutputTerminal source1,
    ScopeFiveVoltsOutOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeFiveVoltsOutOutputTerminal,
    source2 As ScopeFiveVoltsOutOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the first [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the second [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalEquality Operator

Checks whether two [ScopeFiveVoltsOutOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeFiveVoltsOutOutputTerminal source1,
    ScopeFiveVoltsOutOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeFiveVoltsOutOutputTerminal,
    source2 As ScopeFiveVoltsOutOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the first [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the second [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Conversion Operators

Overload List

	Name	Description
	(String to ScopeFiveVoltsOutOutputTerminal)	Converts a specified String to an equivalent ScopeFiveVoltsOutOutputTerminal object.
	(ScopeFiveVoltsOutOutputTerminal to String)	Converts a ScopeFiveVoltsOutOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Conversion (String to ScopeFiveVoltsOutOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeFiveVoltsOutOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeFiveVoltsOutOutputTerminal (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeFiveVoltsOutOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeFiveVoltsOutOutputTerminal](#) object.

Return Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

Returns the [ScopeFiveVoltsOutOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Conversion (String to ScopeFiveVoltsOutOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeFiveVoltsOutOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeFiveVoltsOutOutputTerminal (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeFiveVoltsOutOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeFiveVoltsOutOutputTerminal](#) object.

Return Value

Type: [ScopeFiveVoltsOutOutputTerminal](#)

Returns the [ScopeFiveVoltsOutOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminal Conversion
(ScopeFiveVoltsOutOutputTerminal to String)

Converts a [ScopeFiveVoltsOutOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public static implicit operator string (  
    ScopeFiveVoltsOutOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeFiveVoltsOutOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
Specifies the [ScopeFiveVoltsOutOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeFiveVoltsOutOutputTerminal](#) object.

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFiveVoltsOutOutputTerminalInequality Operator

Checks whether two [ScopeFiveVoltsOutOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeFiveVoltsOutOutputTerminal source1,  
    ScopeFiveVoltsOutOutputTerminal source2  
)
```

```
Public Shared Operator <> (
    source1 As ScopeFiveVoltsOutOutputTerminal,
    source2 As ScopeFiveVoltsOutOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
 Specifies the first [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeFiveVoltsOutOutputTerminal](#)
 Specifies the second [ScopeFiveVoltsOutOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeFiveVoltsOutOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFlexFirAntiAliasFilterType Enumeration

Specifies the type of filter to achieve desired filtering characteristics. The NI 5922 flexible-resolution digitizer uses an onboard FIR lowpass antialias filter.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeFlexFirAntiAliasFilterType
```

```
Public Enumeration ScopeFlexFirAntiAliasFilterType
```

Members

Member name	Value	Description
Tap48Standard	0	<p>This filter is optimized for alias protection and frequency-domain flatness.</p> <ul style="list-style-type: none"> ▪ Alias protection: ranges from 80 dB to 100 dB depending on sample rate ▪ Settling time: within 14 samples from a 50% vertical trigger point ▪ Cutoff frequency: 0.43 x sample rate ▪ Flatness: ripple ranges from 0.005 dB to 0.120 dB depending on selected sample rate. Refer to the NI PXI/PCI-5922 Specifications for more information. ▪ Rise time: approximately $0.75 / (\text{sample rate})$
Tap48Hanning	1	<p>This filter is optimized for the lowest possible bandwidth for a 48-tap filter and maximizes the signal-to-noise ratio (SNR).</p> <ul style="list-style-type: none"> ▪ Settling time: within 14 samples from a 50% vertical trigger point ▪ Cutoff frequency: 0.030 x sample rate ▪ Flatness: 0 to -3dB within cutoff frequency ▪ Rise time: $11.6 / (\text{sample rate})$
Tap16Hanning	2	<p>This filter is optimized for the lowest possible bandwidth for a 16-tap filter and maximizes the SNR.</p> <ul style="list-style-type: none"> ▪ Settling time: 6 samples from a 50% vertical trigger point. ▪ Cutoff frequency: 0.08 x sample rate ▪ Flatness: 0 to -3dB within cutoff frequency

		<ul style="list-style-type: none"> ▪ Rise time: 7.7 / (sample rate)
Tap8Hanning	3	<p>This filter is optimized for the lowest possible bandwidth for an 8-tap filter and maximizes the SNR.</p> <ul style="list-style-type: none"> ▪ Settling time: 4 samples from a 50% vertical trigger point ▪ Cutoff frequency: 0.15 x sample rate ▪ Flatness: 0 to -3 dB within cutoff frequency ▪ Rise time: 3.9 / (sample rate)

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResample Class

Provides configuration options for fractional resampling.

Inheritance Hierarchy

SystemObject

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeFractionalResample

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public sealed class ScopeFractionalResample : ScopeSubObject
```

```
Public NotInheritable Class ScopeFractionalResample
    Inherits ScopeSubObject
```





The ScopeFractionalResample type exposes the following members.

Properties

Name	Description
 Enabled	Gets or sets whether fractional resampling is enabled on the instrument.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

You can use this class to resample the input waveform to the desired sample rate. By using the fractional resampling stage of the onboard signal processing (OSP) block, you can move the fractional resampling processing to hardware without using an external clock.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResample Properties

The [ScopeFractionalResample](#) type exposes the following members.

Properties

Name	Description
 Enabled	Gets or sets whether fractional resampling is enabled on the instrument.

[Top](#)

See Also

Reference

[ScopeFractionalResample Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResampleEnabled Property

Gets or sets whether fractional resampling is enabled on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeFractionalResample Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResampleEnabled Property

Gets or sets whether fractional resampling is enabled on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```


Property Value

Type: [Boolean](#)

The default value is .

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeFractionalResample Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResample Properties

The [ScopeFractionalResample](#) type exposes the following members.

Properties

Name	Description
 Enabled	Gets or sets whether fractional resampling is enabled on the instrument.

[Top](#)

See Also

Reference

[ScopeFractionalResample Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResampleEnabled Property

Gets or sets whether fractional resampling is enabled on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeFractionalResample Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResampleEnabled Property

Gets or sets whether fractional resampling is enabled on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Enabled { get; set; }
```

```
Public Property Enabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is .

Remarks

This property is valid only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. On an instrument that does not support OSP, accessing this property results in a `ModularInstrumentsException`.

See Also

Reference

[ScopeFractionalResample Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeFractionalResample Methods

The [ScopeFractionalResample](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeFractionalResample Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTrigger Class

Represents the sub-object that contains methods and properties used to configure common properties for glitch triggering.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeGlitchTrigger

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB

```
public sealed class ScopeGlitchTrigger : ScopeSubObject,
    IIVI_ScopeTriggerGlitch
```

```
Public NotInheritable Class ScopeGlitchTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTriggerGlitch
```





The ScopeGlitchTrigger type exposes the following members.



Properties

Name	Description
 GlitchCondition	Gets or sets the glitch condition you want the digitizer to use for glitch triggering.
 GlitchPolarity	Gets or sets the glitch polarity you want the digitizer to use for glitch triggering.
 GlitchWidth	Gets or sets the glitch width you want the digitizer to use for glitch triggering.

Top

Methods

Name	Description
 Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)	Configures common properties for analog glitch triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog glitch triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)

 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger related topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTrigger Properties

The [ScopeGlitchTrigger](#) type exposes the following members.

Properties

Name	Description
 GlitchCondition	Gets or sets the glitch condition you want the digitizer to use for glitch triggering.

 GlitchPolarity	Gets or sets the glitch polarity you want the digitizer to use for glitch triggering.
 GlitchWidth	Gets or sets the glitch width you want the digitizer to use for glitch triggering.

[Top](#)

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchCondition Property

Gets or sets the glitch condition you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public ScopeGlitchTriggerCondition GlitchCondition { get; set; }
```

```
Public Property GlitchCondition As ScopeGlitchTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeGlitchTriggerCondition](#)

The glitch condition you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	GlitchCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for glitch condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers either when the glitch width is great than or less than the set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchCondition Property

Gets or sets the glitch condition you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeGlitchTriggerCondition GlitchCondition { get; set; }
```

```
Public Property GlitchCondition As ScopeGlitchTriggerCondition
    Get
    Set
```


Property Value

Type: [ScopeGlitchTriggerCondition](#)

The glitch condition you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers either when the glitch width is great than or less than the set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchPolarity Property

Gets or sets the glitch polarity you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public ScopeTriggerPolarity GlitchPolarity { get; set; }
```

```
Public Property GlitchPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The glitch polarity you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when glitch is positive, negative or either.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchWidth Property

Gets or sets the glitch width you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan GlitchWidth { get; set; }
```

```
Public Property GlitchWidth As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The glitch width you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchWidth was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch width was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width of the glitch is greater than or less than this set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTrigger Properties

The [ScopeGlitchTrigger](#) type exposes the following members.

Properties

Name	Description
 GlitchCondition	Gets or sets the glitch condition you want the digitizer to use for glitch triggering.
 GlitchPolarity	Gets or sets the glitch polarity you want the digitizer to use for glitch triggering.
 GlitchWidth	Gets or sets the glitch width you want the digitizer to use for glitch triggering.

[Top](#)

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchCondition Property

Gets or sets the glitch condition you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeGlitchTriggerCondition GlitchCondition { get; set; }
```

```
Public Property GlitchCondition As ScopeGlitchTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeGlitchTriggerCondition](#)

The glitch condition you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers either when the glitch width is great than or less than the set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchCondition Property

Gets or sets the glitch condition you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeGlitchTriggerCondition GlitchCondition { get; set; }
```

```
Public Property GlitchCondition As ScopeGlitchTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeGlitchTriggerCondition](#)

The glitch condition you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers either when the glitch width is great than or less than the set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchPolarity Property

Gets or sets the glitch polarity you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerPolarity GlitchPolarity { get; set; }
```

```
Public Property GlitchPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The glitch polarity you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when glitch is positive, negative or either.

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerGlitchWidth Property

Gets or sets the glitch width you want the digitizer to use for glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan GlitchWidth { get; set; }
```

```
Public Property GlitchWidth As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The glitch width you want the digitizer to use for glitch triggering.

Exceptions

Exception	Condition
ObjectDisposedException	GlitchWidth was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for glitch width was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width of the glitch is greater than or less than this set width.

See Also

Reference

[ScopeGlitchTrigger Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTrigger Methods

The [ScopeGlitchTrigger](#) type exposes the following members.

Methods

Name	Description
 Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)	Configures common properties for analog glitch triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog glitch triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference



[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method

Overload List

Name	Description
 Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)	Configures common properties for analog glitch triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog glitch triggering.

[Top](#)

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    string source,
    double level,
    PrecisionTimeSpan width,
    Polarity polarity,
    GlitchCondition condition
)
```

```
Public Sub Configure (
    source As String,
    level As Double,
    width As PrecisionTimeSpan,
    polarity As Polarity,
    condition As GlitchCondition
)
```

Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the glitch will trigger.

width

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the width at which the glitch will trigger.

polarity

Type: [Ivi.ScopePolarity](#)

The polarity that the glitch will trigger, either positive, negative, or either.

condition

Type: Ivi.ScopeGlitchCondition

The condition for the glitch to trigger being greater than or less than.

Implements

Ivi.ScopeTriggerGlitchConfigure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition) was called or accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for width was invalid.</p> <p>- or -</p> <p>The value for polarity was invalid.</p> <p>- or -</p> <p>The value for condition was invalid.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    string source,  
    double level,  
    PrecisionTimeSpan width,  
    Polarity polarity,  
    GlitchCondition condition  
)
```

```
Public Sub Configure (  
    source As String,  
    level As Double,  
    width As PrecisionTimeSpan,  
    polarity As Polarity,  
    condition As GlitchCondition  
)
```

Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the glitch will trigger.

width

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the width at which the glitch will trigger.

polarity

Type: [Ivi.ScopePolarity](#)

The polarity that the glitch will trigger, either positive, negative, or either.

condition

Type: [Ivi.ScopeGlitchCondition](#)

The condition for the glitch to trigger being greater than or less than.

Implements

[IviScopeTriggerGlitchConfigure](#)(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Exceptions

Exception	Condition
ObjectDisposedException	Configure (String, Double, PrecisionTimeSpan, Polarity, GlitchCondition) was called or accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	<p>The value for width was invalid.</p> <p>- or -</p> <p>The value for polarity was invalid.</p> <p>- or -</p> <p>The value for condition was invalid.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)
[Configure Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    double level,  
    double glitchWidth,  
    ScopeTriggerPolarity triggerPolarity,  
    ScopeGlitchTriggerCondition glitchCondition,  
    ScopeTriggerCoupling triggerCoupling,  
    PrecisionTimeSpan triggerHoldoff,  
    PrecisionTimeSpan triggerDelay  
)
```

```
Public Sub Configure (  
    triggerSource As ScopeTriggerSource,  
    level As Double,
```

```

    glitchWidth As Double,
    triggerPolarity As ScopeTriggerPolarity,
    glitchCondition As ScopeGlitchTriggerCondition,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the glitch will trigger.

glitchWidth

Type: [SystemDouble](#)

A value indicating the width at which the glitch will trigger.

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)

The polarity that the glitch will trigger, either positive, negative, or either.

glitchCondition

Type: [NationalInstruments.ModularInstruments.NIScopeScopeGlitchTriggerCondition](#)

The condition for the glitch to trigger being greater than or less than.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure(ScopeTriggerSource , Double, Double, ScopeTriggerPolarity , ScopeGlitchTriggerCondition , ScopeTriggerCoupling , PrecisionTimeSpan , PrecisionTimeSpan) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	<p>The value for glitch width was invalid. - or -</p> <p>The value for glitch polarity was invalid. - or -</p> <p>The value for glitch condition was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeGlitchTriggerConfigure Method](#)

Overload List

Name	Description
Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)	Configures common properties for analog glitch triggering.
Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog glitch triggering.

[Top](#)

See Also

Reference

[ScopeGlitchTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public void Configure(
    string source,
    double level,
    PrecisionTimeSpan width,
```

```

    Polarity polarity,
    GlitchCondition condition
)

```

```

Public Sub Configure (
    source As String,
    level As Double,
    width As PrecisionTimeSpan,
    polarity As Polarity,
    condition As GlitchCondition
)

```

Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the glitch will trigger.

width

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the width at which the glitch will trigger.

polarity

Type: [Ivi.ScopePolarity](#)

The polarity that the glitch will trigger, either positive, negative, or either.

condition

Type: [Ivi.ScopeGlitchCondition](#)

The condition for the glitch to trigger being greater than or less than.

Implements

[IviScopeTriggerGlitchConfigure\(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition\)](#)

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition) was called or accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for width was invalid. - or -</p> <p>The value for polarity was invalid. - or -</p> <p>The value for condition was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (String, Double, PrecisionTimeSpan, Polarity, GlitchCondition)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    string source,
    double level,
    PrecisionTimeSpan width,
    Polarity polarity,
```

```

        GlitchCondition condition
    )

```

```

Public Sub Configure (
    source As String,
    level As Double,
    width As PrecisionTimeSpan,
    polarity As Polarity,
    condition As GlitchCondition
)

```

Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the glitch will trigger.

width

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the width at which the glitch will trigger.

polarity

Type: [Ivi.ScopePolarity](#)

The polarity that the glitch will trigger, either positive, negative, or either.

condition

Type: [Ivi.ScopeGlitchCondition](#)

The condition for the glitch to trigger being greater than or less than.

Implements

[Ivi.ScopeTriggerGlitchConfigure\(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition\)](#)

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, Polarity, GlitchCondition) was called or accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for width was invalid. - or -</p> <p>The value for polarity was invalid. - or -</p> <p>The value for condition was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerConfigure Method (ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog glitch triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void Configure(
    ScopeTriggerSource triggerSource,
    double level,
    double glitchWidth,
```

```

ScopeTriggerPolarity triggerPolarity,
ScopeGlitchTriggerCondition glitchCondition,
ScopeTriggerCoupling triggerCoupling,
PrecisionTimeSpan triggerHoldoff,
PrecisionTimeSpan triggerDelay
)

```

```

Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    level As Double,
    glitchWidth As Double,
    triggerPolarity As ScopeTriggerPolarity,
    glitchCondition As ScopeGlitchTriggerCondition,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
The trigger source.

level

Type: [SystemDouble](#)
A value indicating the voltage level at which the glitch will trigger.

glitchWidth

Type: [SystemDouble](#)
A value indicating the width at which the glitch will trigger.

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)
The polarity that the glitch will trigger, either positive, negative, or either.

glitchCondition

Type: [NationalInstruments.ModularInstruments.NIScopeScopeGlitchTriggerCondition](#)
The condition for the glitch to trigger being greater than or less than.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeGlitchTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for glitch width was invalid. - or -</p> <p>The value for glitch polarity was invalid. - or -</p> <p>The value for glitch condition was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeGlitchTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeGlitchTriggerCondition Enumeration

Specifies the polarity of the glitch that triggers the oscilloscope.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeGlitchTriggerCondition
```

```
Public Enumeration ScopeGlitchTriggerCondition
```

Members

Member name	Value	Description
LessThan	1	Glitch will trigger when it is less than the set width
GreaterThan	2	Glitch will trigger when it is greater than the set width.

Remarks

The end-user specifies the width and condition for the glitch, [GlitchWidthGlitchCondition](#)

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Class

Represents the source terminal for [ReferenceClockSource](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax






[C#](#)/[VB](#)





```
public sealed class ScopeInputClockSource
```

```
Public NotInheritable Class ScopeInputClockSource
```

The ScopeInputClockSource type exposes the following members.







Properties

	Name	Description
	Aux0ClkIn	Gets the source terminal for AUX I/O connector 0 Clk In.
	ClockIn	Gets the source terminal for CLK IN.
	External	Gets the source terminal for External.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.

 S	Pfi1	Gets the source terminal for PFI 1.
 S	Pfi2	Gets the source terminal for PFI 2.
 S	PxiClock	Gets the source terminal for PXI CLK.
 S	RtsiClock	Gets the source terminal for RTSI Clock.




[Top](#)


Methods

	Name	Description
 S	Equals(Object)	Determines whether the current instance of <code>ScopeInputClockSource</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 S	Equals(ScopeInputClockSource)	Determines whether the current instance of <code>ScopeInputClockSource</code> and the specified <code>ScopeInputClockSource</code> object are equal.
 S	FromString	Creates a <code>ScopeInputClockSource</code> object from the specified <code>String</code> .
 S	GetHashCode	Gets the hash code for the current instance of <code>ScopeInputClockSource</code> . (Overrides ObjectGetHashCode .)
 S	GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 S	ToString	Converts the current instance of <code>ScopeInputClockSource</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 S	Equality	Checks whether two <code>ScopeInputClockSource</code> object instances are equal.
 S	(String to ScopeInputClockSource)	Converts a specified <code>String</code> to an equivalent <code>ScopeInputClockSource</code> object.
 S	(ScopeInputClockSource to String)	Converts a <code>ScopeInputClockSource</code> object to an equivalent <code>String</code> .

	Inequality	Checks whether two ScopeInputClockSource object instances are not equal.
---	----------------------------	--

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeTiming](#)








[ScopeTimingReferenceClockSource](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Properties

The [ScopeInputClockSource](#) type exposes the following members.

Properties

	Name	Description
	Aux0ClkIn	Gets the source terminal for AUX I/O connector 0 Clk In.
	ClockIn	Gets the source terminal for CLK IN.
	External	Gets the source terminal for External.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.

 PxiClock	Gets the source terminal for PXI CLK.
 RtSiClock	Gets the source terminal for RTSI Clock.

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceAux0ClkIn Property

Gets the source terminal for AUX I/O connector 0 Clk In.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static ScopeInputClockSource Aux0ClkIn { get; }
```

```
Public Shared ReadOnly Property Aux0ClkIn As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceAux0ClkIn Property

Gets the source terminal for AUX I/O connector 0 Clk In.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Aux0ClkIn { get; }
```

```
Public Shared ReadOnly Property Aux0ClkIn As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceExternal Property

Gets the source terminal for External.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource External { get; }
```

```
Public Shared ReadOnly Property External As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_EXTERNAL".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePxiClock Property

Gets the source terminal for PXI CLK.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource PxiClock { get; }
```

```
Public Shared ReadOnly Property PxiClock As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PXI_CLK".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceRtsiClock Property

Gets the source terminal for RTSI Clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource RtsiClock { get; }
```

```
Public Shared ReadOnly Property RtsiClock As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_RTISI_CLOCK".

See Also

Reference

[ScopeInputClockSource Class](#)










[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Properties

The [ScopeInputClockSource](#) type exposes the following members.

Properties

	Name	Description
	Aux0ClkIn	Gets the source terminal for AUX I/O connector 0 Clk In.
	ClockIn	Gets the source terminal for CLK IN.
	External	Gets the source terminal for External.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxiClock	Gets the source terminal for PXI CLK.
	RtsiClock	Gets the source terminal for RTSI Clock.

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceAux0ClkIn Property

Gets the source terminal for AUX I/O connector 0 Clk In.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Aux0ClkIn { get; }
```

```
Public Shared ReadOnly Property Aux0ClkIn As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceAux0ClkIn Property

Gets the source terminal for AUX I/O connector 0 Clk In.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Aux0ClkIn { get; }
```

```
Public Shared ReadOnly Property Aux0ClkIn As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceExternal Property

Gets the source terminal for External.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource External { get; }
```

```
Public Shared ReadOnly Property External As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_EXTERNAL".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourcePxiClock Property

Gets the source terminal for PXI CLK.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource PxiClock { get; }
```

```
Public Shared ReadOnly Property PxiClock As ScopeInputClockSource  
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_PXI_CLK".

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceRtSiClock Property

Gets the source terminal for RTSI Clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource RtsiClock { get; }
```

```
Public Shared ReadOnly Property RtsiClock As ScopeInputClockSource
    Get
```

Property Value

Type: [ScopeInputClockSource](#)

A [ScopeInputClockSource](#) representing the [String](#) "VAL_RTISI_CLOCK".

See Also

Reference

[ScopeInputClockSource Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Methods

The [ScopeInputClockSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeInputClockSource and the specified object are equal. (Overrides Object.Equals(Object) .)

 Equals(ScopelInputClockSource)	Determines whether the current instance of ScopelInputClockSource and the specified ScopelInputClockSource object are equal.
 FromString	Creates a ScopelInputClockSource object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopelInputClockSource . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopelInputClockSource to String . (Overrides ObjectToString .)

[Top](#)

See Also



Reference

[ScopelInputClockSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopelInputClockSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopelInputClockSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopelInputClockSource)	Determines whether the current instance of ScopelInputClockSource and the specified ScopelInputClockSource object are equal.

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```


Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (ScopeInputClockSource)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified [ScopeInputClockSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeInputClockSource source
)
```

```
Public Function Equals (
    source As ScopeInputClockSource
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the [ScopeInputClockSource](#) object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)


[Equals Overload](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeInputClockSource and the specified object are equal. (Overrides ObjectEquals(Object) .)

 [Equals\(ScopeInputClockSource\)](#) Determines whether the current instance of [ScopeInputClockSource](#) and the specified [ScopeInputClockSource](#) object are equal.

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquals Method (ScopeInputClockSource)

Determines whether the current instance of [ScopeInputClockSource](#) and the specified [ScopeInputClockSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public bool Equals(  
    ScopeInputClockSource source  
)
```

```
Public Function Equals (  
    source As ScopeInputClockSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the [ScopeInputClockSource](#) object to compare to the current instance of [ScopeInputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceFromString Method

Creates a [ScopeInputClockSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeInputClockSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeInputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeInputClockSource](#) object returned represents.

Return Value

Type: [ScopeInputClockSource](#)

Returns an object of type [ScopeInputClockSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeInputClockSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeInputClockSource](#).

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceToString Method

Converts the current instance of [ScopeInputClockSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```



```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeInputClockSource](#).

See Also

Reference

[ScopeInputClockSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Operators and Type Conversions

The [ScopeInputClockSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeInputClockSource object instances are equal.
	(String to ScopeInputClockSource)	Converts a specified String to an equivalent ScopeInputClockSource object.
	(ScopeInputClockSource to String)	Converts a ScopeInputClockSource object to an equivalent String .
	Inequality	Checks whether two ScopeInputClockSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquality Operator

Checks whether two [ScopeInputClockSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeInputClockSource source1,
    ScopeInputClockSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeInputClockSource,
    source2 As ScopeInputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the first [ScopeInputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the second [ScopeInputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceEquality Operator

Checks whether two [ScopeInputClockSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeInputClockSource source1,
    ScopeInputClockSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeInputClockSource,
    source2 As ScopeInputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the first [ScopeInputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the second [ScopeInputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeInputClockSource)	Converts a specified String to an equivalent ScopeInputClockSource object.
	(ScopeInputClockSource to String)	Converts a ScopeInputClockSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Conversion (String to ScopeInputClockSource)

Converts a specified [String](#) to an equivalent [ScopeInputClockSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeInputClockSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeInputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeInputClockSource](#) object.

Return Value

Type: [ScopeInputClockSource](#)

Returns the [ScopeInputClockSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeInputClockSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Conversion (String to ScopeInputClockSource)

Converts a specified [String](#) to an equivalent [ScopeInputClockSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeInputClockSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeInputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeInputClockSource](#) object.

Return Value

Type: [ScopeInputClockSource](#)

Returns the [ScopeInputClockSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeInputClockSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSource Conversion (ScopeInputClockSource to String)

Converts a [ScopeInputClockSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeInputClockSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeInputClockSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the [ScopeInputClockSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeInputClockSource](#) object.

See Also

Reference

[ScopeInputClockSource Class](#)
[Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInputClockSourceInequality Operator

Checks whether two [ScopeInputClockSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeInputClockSource source1,
    ScopeInputClockSource source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeInputClockSource,
    source2 As ScopeInputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the first [ScopeInputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
Specifies the second [ScopeInputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeInputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgs Class

Provides NI-SCOPE interchange check warning event arguments.

Inheritance Hierarchy

[SystemObject](#)

[SystemEventArgs](#)

NationalInstruments.ModularInstruments.NIScopeScopeInterchangeCheckWarningEventArgs

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeInterchangeCheckWarningEventArgs : EventArgs
```

```
Public NotInheritable Class ScopeInterchangeCheckWarningEventArgs
    Inherits EventArgs
```





The `ScopeInterchangeCheckWarningEventArgs` type exposes the following members.

Properties

	Name	Description
	Text	Gets the interchange check warning message.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to [NI High Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)


Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgs

Properties

The [ScopeInterchangeCheckWarningEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Text	Gets the interchange check warning message.

[Top](#)

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgsText Property

Gets the interchange check warning message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that represent the interchange check warning message.

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgsText Property

Gets the interchange check warning message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that represent the interchange check warning message.

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgs Properties

The [ScopeInterchangeCheckWarningEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Text	Gets the interchange check warning message.

[Top](#)

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgsText Property

Gets the interchange check warning message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) that represent the interchange check warning message.

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgsText Property

Gets the interchange check warning message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) that represent the interchange check warning message.

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeInterchangeCheckWarningEventArgs Methods

The [ScopeInterchangeCheckWarningEventArgs](#) type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeInterchangeCheckWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurement Class

Represents the [NIScope](#) sub-objects that contain the properties and methods related to acquisition.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeMeasurement

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax







C#VB


```
public sealed class ScopeMeasurement : ScopeSubObject,
    IIVI_ScopeMeasurement
```

```
Public NotInheritable Class ScopeMeasurement
    Inherits ScopeSubObject
    Implements IIVI_ScopeMeasurement
```

The ScopeMeasurement type exposes the following members.










Properties

Name	Description
 FetchInterleavedData	Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.
 FetchInterleavedIQData	Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms.
 FetchOffset	Specifies the offset, in samples, to start fetching data within each record. The offset is applied relative to FetchRelativeTo . The offset can be positive or negative.
 FetchRelativeTo	Gets or sets the position to start fetching within one record.
 PointsDone	Gets the actual number of samples acquired since the last fetch, relative to the configured value for FetchRelativeTo , including FetchOffset and for the current configured RecordNumberToFetch .
 RecordsDone	Gets the number of records that have been completely acquired.

 RisInAutoSetupAllowed	Specifies whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.
---	---

[Top](#)

Methods

Name	Description
 Abort	Aborts an acquisition and returns the digitizer to the Idle state. Call this method if the digitizer times out waiting for a trigger.
 AutoSetup	Automatically configures the instrument. When this method is called, the digitizer senses the input signal and automatically configures many of the instrument settings. If a signal is detected on a channel, the driver chooses the smallest available vertical range that is larger than the signal range.
 Commit	Commits to hardware all the parameter settings associated with the task.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 Initiate	Initiates a waveform acquisition.
 Status	Gets the status information about the acquisition.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Acquisition Functions topic in the [NI High Speed Digitizers Help](#) or to the Fetch and Read topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources








[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurement Properties

The [ScopeMeasurement](#) type exposes the following members.

Properties

Name	Description
 FetchInterleavedData	Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.
 FetchInterleavedIQData	Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms.
 FetchOffset	Specifies the offset, in samples, to start fetching data within each record. The offset is applied relative to FetchRelativeTo . The offset can be positive or negative.
 FetchRelativeTo	Gets or sets the position to start fetching within one record.
 PointsDone	Gets the actual number of samples acquired since the last fetch, relative to the configured value for FetchRelativeTo , including FetchOffset and for the current configured RecordNumberToFetch .
 RecordsDone	Gets the number of records that have been completely acquired.
 RisInAutoSetupAllowed	Specifies whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

[Top](#)

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedData Property

Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedData { get; set; }
```

```
Public Property FetchInterleavedData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, to retrieve one array with alternating values on the NI 5620/5621; otherwise, .

Exceptions

Exception	Condition
ObjectDisposedException	FetchInterleavedData was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can use this attribute to retrieve a single array with I and Q interleaved instead of two separate arrays. If set to `true`, the resulting array is twice the size of the actual record length.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedData Property

Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedData { get; set; }
```

```
Public Property FetchInterleavedData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

`true`, to retrieve one array with alternating values on the NI 5620/5621; otherwise, `false`.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInterleavedData was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can use this attribute to retrieve a single array with I and Q interleaved instead of two separate arrays. If set to , the resulting array is twice the size of the actual record length.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedIQData Property

Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedIQData { get; set; }
```

```
Public Property FetchInterleavedIQData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, if the number of elements returned by scalar fetch types (such as 16-bit integer) is twice the requested number of samples, and if during DDC acquisitions in Complex mode, two noninterleaved arrays of data are returned per channel, per record. The default value is .

Remarks

This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE throws an exception if you use this attribute with an instrument that does not support OSP.

Note

If this is set to , then the following methods, as well as their overloads and async versions, will return waveforms with twice the number of samples specified: [Read\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble\)](#), [FetchByte\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte\)](#), [FetchInt16\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16\)](#), [FetchInt32\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32\)](#), and [FetchDouble\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble\)](#).

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchOffset Property

Specifies the offset, in samples, to start fetching data within each record. The offset is applied relative to [FetchRelativeTo](#). The offset can be positive or negative.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long FetchOffset { get; set; }
```

```
Public Property FetchOffset As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value of this property is instrument-dependent.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchRelativeTo Property

Gets or sets the position to start fetching within one record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFetchRelativeTo FetchRelativeTo { get; set; }
```

```
Public Property FetchRelativeTo As ScopeFetchRelativeTo  
    Get  
    Set
```

Property Value

Type: [ScopeFetchRelativeTo](#)

The default value of this property is instrument-dependent. Valid values are [Now](#), [Pretrigger](#), [ReadPointer](#), [Start](#), and [Trigger](#).

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementPointsDone Property

Gets the actual number of samples acquired since the last fetch, relative to the configured value for `FetchRelativeTo`, including `FetchOffset` and for the current configured `RecordNumberToFetch`.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public double PointsDone { get; }
```



```
Public ReadOnly Property PointsDone As Double
    Get
```

Property Value

Type: [Double](#)

Actual number of samples acquired since the last fetch.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementRecordsDone Property

Gets the number of records that have been completely acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordsDone { get; }
```

```
Public ReadOnly Property RecordsDone As Long
    Get
```

Property Value

Type: [Int64](#)

Number of records that have been completely acquired.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementRisInAutoSetupAllowed Property

Specifies whether the digitizer should use RIS sample rates when searching for a frequency in autosetup.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RisInAutoSetupAllowed { get; set; }
```

```
Public Property RisInAutoSetupAllowed As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

Value indicating whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

See Also

Reference

[ScopeMeasurement Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurement Properties

The [ScopeMeasurement](#) type exposes the following members.

Properties

Name	Description
 FetchInterleavedData	Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.
 FetchInterleavedIQData	Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms.
 FetchOffset	Specifies the offset, in samples, to start fetching data within each record. The offset is applied relative to FetchRelativeTo . The offset can be positive or negative.
 FetchRelativeTo	Gets or sets the position to start fetching within one record.
 PointsDone	Gets the actual number of samples acquired since the last fetch, relative to the configured value for FetchRelativeTo , including FetchOffset and for the current configured RecordNumberToFetch .
 RecordsDone	Gets the number of records that have been completely acquired.
 RisInAutoSetupAllowed	Specifies whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

[Top](#)

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedData Property

Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedData { get; set; }
```

```
Public Property FetchInterleavedData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, to retrieve one array with alternating values on the NI 5620/5621; otherwise, .

Exceptions

Exception	Condition
ObjectDisposedException	FetchInterleavedData was accessed after the associated NIScope or ScopeDriverUtility object was disposed.

IviCDriverException

The underlying NI-SCOPE driver returned an error.

Remarks

You can use this attribute to retrieve a single array with I and Q interleaved instead of two separate arrays. If set to `true`, the resulting array is twice the size of the actual record length.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedData Property

Gets or sets a value specifying whether to retrieve one array with alternating values on the NI 5620/5621.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedData { get; set; }
```

```
Public Property FetchInterleavedData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

`true`, to retrieve one array with alternating values on the NI 5620/5621; otherwise, `false`.

Exceptions

Exception	Condition
ObjectDisposedException	FetchInterleavedData was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

You can use this attribute to retrieve a single array with I and Q interleaved instead of two separate arrays. If set to , the resulting array is twice the size of the actual record length.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchInterleavedIQData Property

Specifies whether a fetch call retrieves a single waveform with I and Q interleaved, or two separate waveforms.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool FetchInterleavedIQData { get; set; }
```

```
Public Property FetchInterleavedIQData As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

, if the number of elements returned by scalar fetch types (such as 16-bit integer) is twice the requested number of samples, and if during DDC acquisitions in Complex mode, two noninterleaved arrays of data are returned per channel, per record. The default value is .

Remarks

This attribute can be used only with high-speed digitizers that support onboard signal processing (OSP). NI-SCOPE throws an exception if you use this attribute with an instrument that does not support OSP.

Note

If this is set to , then the following methods, as well as their overloads and async versions, will return waveforms with twice the number of samples specified: [Read\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble\)](#), [FetchByte\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionByte\)](#), [FetchInt16\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt16\)](#), [FetchInt32\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionInt32\)](#), and [FetchDouble\(PrecisionTimeSpan, Int64, AnalogWaveformCollectionDouble\)](#).

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchOffset Property

Specifies the offset, in samples, to start fetching data within each record. The offset is applied relative to [FetchRelativeTo](#). The offset can be positive or negative.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long FetchOffset { get; set; }
```

```
Public Property FetchOffset As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value of this property is instrument-dependent.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFetchRelativeTo Property

Gets or sets the position to start fetching within one record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFetchRelativeTo FetchRelativeTo { get; set; }
```

```
Public Property FetchRelativeTo As ScopeFetchRelativeTo
    Get
    Set
```


Property Value

Type: [ScopeFetchRelativeTo](#)

The default value of this property is instrument-dependent. Valid values are [Now](#), [Pretrigger](#), [ReadPointer](#), [Start](#), and [Trigger](#).

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementPointsDone Property

Gets the actual number of samples acquired since the last fetch, relative to the configured value for `FetchRelativeTo`, including `FetchOffset` and for the current configured `RecordNumberToFetch`.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public double PointsDone { get; }
```

```
Public ReadOnly Property PointsDone As Double  
    Get
```

Property Value

Type: [Double](#)

Actual number of samples acquired since the last fetch.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementRecordsDone Property

Gets the number of records that have been completely acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RecordsDone { get; }
```

```
Public ReadOnly Property RecordsDone As Long  
    Get
```

Property Value

Type: [Int64](#)

Number of records that have been completely acquired.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementRisInAutoSetupAllowed Property

Specifies whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool RisInAutoSetupAllowed { get; set; }
```

```
Public Property RisInAutoSetupAllowed As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

Value indicating whether the digitizer should use RIS sample rates when searching for a frequency in autoseup.

See Also

Reference

[ScopeMeasurement Class](#)










[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurement Methods

The [ScopeMeasurement](#) type exposes the following members.

Methods

Name	Description
 Abort	Aborts an acquisition and returns the digitizer to the Idle state. Call this method if the digitizer times out waiting for a trigger.
 AutoSetup	Automatically configures the instrument. When this method is called, the digitizer senses the input signal and automatically configures many of the instrument settings. If a signal is detected on a channel, the driver chooses the smallest available vertical range that is larger than the signal range.
 Commit	Commits to hardware all the parameter settings associated with the task.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 Initiate	Initiates a waveform acquisition.
 Status	Gets the status information about the acquisition.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

[See Also](#)

[Reference](#)

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementAbort Method

Aborts an acquisition and returns the digitizer to the Idle state. Call this method if the digitizer times out waiting for a trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Abort()
```

```
Public Sub Abort
```

Implements

IlviScopeMeasurementAbort

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementAbort Method

Aborts an acquisition and returns the digitizer to the Idle state. Call this method if the digitizer times out waiting for a trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Abort()
```

```
Public Sub Abort
```

Implements

[IIVI_ScopeMeasurementAbort](#)

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementAutoSetup Method

Automatically configures the instrument. When this method is called, the digitizer senses the input signal and automatically configures many of the instrument settings. If a signal is detected on a channel, the driver chooses the smallest available vertical range that is larger than the signal range.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void AutoSetup()
```

```
Public Sub AutoSetup
```

Implements

IlviScopeMeasurementAutoSetup

Remarks

If no signal is found on any analog input channel, a warning is returned, and all channels are enabled. A channel is considered to have a signal present if the signal is at least 10% of the smallest vertical range available for that channel.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementCommit Method

Commits to hardware all the parameter settings associated with the task.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Commit()
```

```
Public Sub Commit
```

Remarks

Use this method if you want a parameter change to be immediately reflected in the hardware. This method is not supported for Traditional NI-DAQ (Legacy) instruments.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementInitiate Method

Initiates a waveform acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Initiate()
```

```
Public Sub Initiate
```

Implements

IIviScopeMeasurementInitiate

Remarks

After you call this method, the digitizer leaves the Idle state and waits for a trigger. The digitizer acquires a waveform for each enabled channel.

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementStatus Method

Gets the status information about the acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAcquisitionStatus Status()
```

```
Public Function Status As ScopeAcquisitionStatus
```

Field Value

Type: [ScopeAcquisitionStatus](#)

The [ScopeAcquisitionStatus](#).

Return Value

Type: [ScopeAcquisitionStatus](#)

Returns [ScopeAcquisitionStatus](#).

See Also

Reference

[ScopeMeasurement Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFilterType Enumeration

Specifies the type of filter, for both IIR and FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeMeasurementFilterType
```

```
Public Enumeration ScopeMeasurementFilterType
```

Members

Member name	Value	Description
LowPass	0	A Lowpass filter value.
HighPass	1	A Highpass filter.
BandPass	2	A Bandpass filter.

BandStop	3	A Bandstop filter.
----------	---	--------------------

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementFirFilterWindow Enumeration

Specifies the FIR window type. The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeMeasurementFirFilterWindow
```

```
Public Enumeration ScopeMeasurementFirFilterWindow
```

Members

Member name	Value	Description
None	0	Windowing is not applied to the FIR filter coefficients.
HanningWindow	409	A Hanning window is applied to the FIR filter coefficients.
FlatTopWindow	410	A Flat Top window is applied to the FIR filter coefficients.
HammingWindow	420	A Hamming window is applied to the FIR filter coefficients.
TriangleWindow	423	A Triangle window is applied to the FIR filter coefficients.
BlackmanWindow	424	A Blackman window is applied to the FIR filter coefficients.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementPercentageMethod Enumeration

Specifies the method used to map percentage reference units to voltages for the reference.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeMeasurementPercentageMethod
```

```
Public Enumeration ScopeMeasurementPercentageMethod
```

Members

Member name	Value	Description
LowHigh	0	The reference level percentages must be computed using the low/high method.
MinMax	1	The reference level percentages must be computed using the min/max method.
BaseTop	2	The reference level percentages must be computed using the base-to-top method.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeMeasurementReferenceLevelUnits](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeMeasurementReferenceLevelUnits Enumeration

Specifies the units of the waveform measurement reference levels. If you choose Percentage, then the measurement routine uses [PercentageMethod](#) to map the percentage values to voltages. By choosing Voltage, you can set the voltage threshold values directly and avoid extra calculations.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeMeasurementReferenceLevelUnits
```

```
Public Enumeration ScopeMeasurementReferenceLevelUnits
```

Members

Member name	Value	Description
Voltage	0	The reference levels are given in units of volts.
Percentage	1	The reference levels are given as a percentage, where the measurements voltage low and voltage high represent 0% and 100%, respectively.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeMeasurementPercentageMethod](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Class

Represents the output terminal for [ReferenceClockOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax












C#VB





```
public sealed class ScopeOutputClockSource
```

```
Public NotInheritable Class ScopeOutputClockSource
```

The ScopeOutputClockSource type exposes the following members.








Properties

	Name	Description
	Aux0ClkOut	Gets the output terminal for AUX I/O connector 0 Clk Out.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.

 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.
 RtsiClock	Gets the output terminal for RTSI Clock.







[Top](#)


Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeOutputClockSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeOutputClockSource)	Determines whether the current instance of ScopeOutputClockSource and the specified ScopeOutputClockSource object are equal.
	 FromString	Creates a ScopeOutputClockSource object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeOutputClockSource. (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeOutputClockSource to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two ScopeOutputClockSource object instances are equal.
 	(String to ScopeOutputClockSource)	Converts a specified String to an equivalent ScopeOutputClockSource object.
 	(ScopeOutputClockSource to String)	Converts a ScopeOutputClockSource object to an equivalent String .

	Inequality	Checks whether two ScopeOutputClockSource object instances are not equal.
---	----------------------------	---

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)








[ScopeExportedSignalsReferenceClockOutputTerminal](#)









Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Properties

The [ScopeOutputClockSource](#) type exposes the following members.

Properties

	Name	Description
	Aux0ClkOut	Gets the output terminal for AUX I/O connector 0 Clk Out.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	Rtsi0	Gets the output terminal for RTSI 0.

 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.
 RtsiClock	Gets the output terminal for RTSI Clock.

[Top](#)

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceAux0ClkOut Property

Gets the output terminal for AUX I/O connector 0 Clk Out.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static ScopeOutputClockSource Aux0ClkOut { get; }
```

```
Public Shared ReadOnly Property Aux0ClkOut As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceAux0ClkOut Property

Gets the output terminal for AUX I/O connector 0 Clk Out.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Aux0ClkOut { get; }
```

```
Public Shared ReadOnly Property Aux0ClkOut As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi2 { get; }
```



```
Public Shared ReadOnly Property Rtsi2 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsiClock Property

Gets the output terminal for RTSI Clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource RtsiClock { get; }
```

```
Public Shared ReadOnly Property RtsiClock As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_CLOCK".

See Also

Reference

[ScopeOutputClockSource Class](#)
















[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Properties

The [ScopeOutputClockSource](#) type exposes the following members.

Properties

	Name	Description
	Aux0ClkOut	Gets the output terminal for AUX I/O connector 0 Clk Out.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.
	RtsiClock	Gets the output terminal for RTSI Clock.

[Top](#)

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceAux0ClkOut Property

Gets the output terminal for AUX I/O connector 0 Clk Out.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Aux0ClkOut { get; }
```

```
Public Shared ReadOnly Property Aux0ClkOut As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceAux0ClkOut Property

Gets the output terminal for AUX I/O connector 0 Clk Out.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Aux0ClkOut { get; }
```

```
Public Shared ReadOnly Property Aux0ClkOut As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_AUX_0_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource ClockOut { get; }
```



```
Public Shared ReadOnly Property ClockOut As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourcePfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeOutputClockSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeOutputClockSource  
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceRtsiClock Property

Gets the output terminal for RTSI Clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeOutputClockSource RtsiClock { get; }
```

```
Public Shared ReadOnly Property RtsiClock As ScopeOutputClockSource
    Get
```

Property Value

Type: [ScopeOutputClockSource](#)

A [ScopeOutputClockSource](#) representing the [String](#) "VAL_RTSI_CLOCK".

See Also

Reference

[ScopeOutputClockSource Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Methods

The [ScopeOutputClockSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeOutputClockSource and the specified object are equal. (Overrides Object.Equals(Object) .)
	Equals(ScopeOutputClockSource)	Determines whether the current instance of ScopeOutputClockSource and the specified ScopeOutputClockSource object are equal.
	FromString	Creates a ScopeOutputClockSource object from the specified String .



 GetHashCode	Gets the hash code for the current instance of ScopeOutputClockSource . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeOutputClockSource to String . (Overrides ObjectToString .)

[Top](#)[See Also](#)[Reference](#)[ScopeOutputClockSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method

[Overload List](#)

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeOutputClockSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeOutputClockSource)	Determines whether the current instance of ScopeOutputClockSource and the specified ScopeOutputClockSource object are equal.

[Top](#)[See Also](#)[Reference](#)[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (ScopeOutputClockSource)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified [ScopeOutputClockSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeOutputClockSource source  
)
```

```
Public Function Equals (  
    source As ScopeOutputClockSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
 Specifies the [ScopeOutputClockSource](#) object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeOutputClockSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeOutputClockSource)	Determines whether the current instance of ScopeOutputClockSource and the specified ScopeOutputClockSource object are equal.

[Top](#)

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (Object)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquals Method (ScopeOutputClockSource)

Determines whether the current instance of [ScopeOutputClockSource](#) and the specified [ScopeOutputClockSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeOutputClockSource source  
)
```

```
Public Function Equals (  
    source As ScopeOutputClockSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the [ScopeOutputClockSource](#) object to compare to the current instance of [ScopeOutputClockSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceFromString Method

Creates a [ScopeOutputClockSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static ScopeOutputClockSource FromString(
    string source
)
```

```
Public Shared Function FromString (
    source As String
) As ScopeOutputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeOutputClockSource](#) object returned represents.

Return Value

Type: [ScopeOutputClockSource](#)

Returns an object of type [ScopeOutputClockSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeOutputClockSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeOutputClockSource](#).

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceToString Method

Converts the current instance of [ScopeOutputClockSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeOutputClockSource](#).

See Also

Reference

[ScopeOutputClockSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Operators and Type Conversions

The [ScopeOutputClockSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeOutputClockSource object instances are equal.
	(String to ScopeOutputClockSource)	Converts a specified String to an equivalent ScopeOutputClockSource object.
	(ScopeOutputClockSource to String)	Converts a ScopeOutputClockSource object to an equivalent String .
	Inequality	Checks whether two ScopeOutputClockSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquality Operator

Checks whether two [ScopeOutputClockSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeOutputClockSource source1,
    ScopeOutputClockSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeOutputClockSource,
    source2 As ScopeOutputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the first [ScopeOutputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the second [ScopeOutputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceEquality Operator

Checks whether two [ScopeOutputClockSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeOutputClockSource source1,
    ScopeOutputClockSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeOutputClockSource,
    source2 As ScopeOutputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the first [ScopeOutputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the second [ScopeOutputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeOutputClockSource)	Converts a specified String to an equivalent ScopeOutputClockSource object.
	(ScopeOutputClockSource to String)	Converts a ScopeOutputClockSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Conversion (String to ScopeOutputClockSource)

Converts a specified [String](#) to an equivalent [ScopeOutputClockSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeOutputClockSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeOutputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeOutputClockSource](#) object.

Return Value

Type: [ScopeOutputClockSource](#)

Returns the [ScopeOutputClockSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeOutputClockSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Conversion (String to ScopeOutputClockSource)

Converts a specified [String](#) to an equivalent [ScopeOutputClockSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeOutputClockSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeOutputClockSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeOutputClockSource](#) object.

Return Value

Type: [ScopeOutputClockSource](#)

Returns the [ScopeOutputClockSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeOutputClockSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSource Conversion (ScopeOutputClockSource to String)

Converts a [ScopeOutputClockSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeOutputClockSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeOutputClockSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the [ScopeOutputClockSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeOutputClockSource](#) object.

See Also

Reference

[ScopeOutputClockSource Class](#)
[Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOutputClockSourceInequality Operator

Checks whether two [ScopeOutputClockSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeOutputClockSource source1,
    ScopeOutputClockSource source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeOutputClockSource,
    source2 As ScopeOutputClockSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the first [ScopeOutputClockSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
Specifies the second [ScopeOutputClockSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeOutputClockSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeOverflowErrorReportingMode Enumeration

Controls the overflow error reporting behavior of the digital downconverter (DDC) block of the onboard signal processing (OSP) device.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeOverflowErrorReportingMode
```

```
Public Enumeration ScopeOverflowErrorReportingMode
```

Members

Member name	Value	Description
Error	0	A ModularInstrumentsException is thrown when an overflow occurs in the onboard signal processing (OSP) block.
Warning	1	No exceptions are thrown when an overflow occurs in the onboard signal processing (OSP) block. A warning occurs. Hook onto Warning of ModularInstrumentsWarning to get the warning message.
Disabled	2	No exceptions are thrown when an overflow occurs in the onboard signal processing (OSP) block.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[ScopeAcquisitionOverflowErrorReportingMode](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Class

Represents the output terminal for [ReadyForAdvanceEventOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB




















```
public sealed class ScopeReadyForAdvanceEventOutputTerminal
```

```
Public NotInheritable Class ScopeReadyForAdvanceEventOutputTerminal
```

The `ScopeReadyForAdvanceEventOutputTerminal` type exposes the following members.



Properties






	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.

 Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
 Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
 Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
 Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
 Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
 ClockOut	Gets the output terminal for CLK OUT.
 NoSource	Gets the output terminal for no source.
 Pfi0	Gets the output terminal for PFI 0.
 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)









Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeReadyForAdvanceEventOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForAdvanceEventOutputTerminal)	Determines whether the current instance of <code>ScopeReadyForAdvanceEventOutputTerminal</code> and the specified <code>ScopeReadyForAdvanceEventOutputTerminal</code> object are equal.

 FromString 	Creates a <code>ScopeReadyForAdvanceEventOutputTerminal</code> object from the specified String .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeReadyForAdvanceEventOutputTerminal</code> . (Overrides Object.GetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeReadyForAdvanceEventOutputTerminal</code> to String . (Overrides Object.ToString .)

[Top](#)

Operators

Name	Description
 Equality 	Checks whether two <code>ScopeReadyForAdvanceEventOutputTerminal</code> object instances are equal.
 (String to ScopeReadyForAdvanceEventOutputTerminal) 	Converts a specified String to an equivalent <code>ScopeReadyForAdvanceEventOutputTerminal</code> object.
 (ScopeReadyForAdvanceEventOutputTerminal to String) 	Converts a <code>ScopeReadyForAdvanceEventOutputTerminal</code> object to an equivalent String .
 Inequality 	Checks whether two <code>ScopeReadyForAdvanceEventOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference












[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsReadyForAdvanceEventOutputTerminal](#)












Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Properties

The [ScopeReadyForAdvanceEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.

 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi2

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)
 A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As
    ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As
    ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalClockOut

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi0 Pr

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)
A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi1 Pr

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi2 Pr

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPxiStar

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi0

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi1

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi2

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi3 P

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi4 P

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi5

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi6 P

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi7

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)























[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Properties

The [ScopeReadyForAdvanceEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReadyForAdvanceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```


Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
    ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As
ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)
 A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeReadyForAdvanceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeReadyForAdvanceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

A [ScopeReadyForAdvanceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference








[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Methods

The [ScopeReadyForAdvanceEventOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForAdvanceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified ScopeReadyForAdvanceEventOutputTerminal object are equal.
 FromString 	Creates a ScopeReadyForAdvanceEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeReadyForAdvanceEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeReadyForAdvanceEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForAdvanceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified ScopeReadyForAdvanceEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified [ScopeReadyForAdvanceEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReadyForAdvanceEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReadyForAdvanceEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForAdvanceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForAdvanceEventOutputTerminal and the specified ScopeReadyForAdvanceEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeReadyForAdvanceEventOutputTerminalEquals Method](#)
([ScopeReadyForAdvanceEventOutputTerminal](#))

Determines whether the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) and the specified [ScopeReadyForAdvanceEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReadyForAdvanceEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReadyForAdvanceEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeReadyForAdvanceEventOutputTerminalFromString Method](#)

Creates a [ScopeReadyForAdvanceEventOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForAdvanceEventOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeReadyForAdvanceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeReadyForAdvanceEventOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

Returns an object of type [ScopeReadyForAdvanceEventOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalGetHashCode Method

Gets the hash code for the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalToString Method

Converts the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeReadyForAdvanceEventOutputTerminal](#).

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Operators and Type Conversions

The [ScopeReadyForAdvanceEventOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeReadyForAdvanceEventOutputTerminal object instances are equal.
	(String to ScopeReadyForAdvanceEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForAdvanceEventOutputTerminal object.

 	(ScopeReadyForAdvanceEventOutputTerminal to String)	Converts a ScopeReadyForAdvanceEventOutputTerminal object to an equivalent String .
 	Inequality	Checks whether two ScopeReadyForAdvanceEventOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquality

Checks whether two [ScopeReadyForAdvanceEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReadyForAdvanceEventOutputTerminal source1,
    ScopeReadyForAdvanceEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReadyForAdvanceEventOutputTerminal,
    source2 As ScopeReadyForAdvanceEventOutputTerminal
) As Boolean
```


Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the first [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the second [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalEquality Operator

Checks whether two [ScopeReadyForAdvanceEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReadyForAdvanceEventOutputTerminal source1,
```

```
ScopeReadyForAdvanceEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReadyForAdvanceEventOutputTerminal,
    source2 As ScopeReadyForAdvanceEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
 Specifies the first [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
 Specifies the second [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Conversion Operators

Overload List

Name	Description
------	-------------

 	(String to ScopeReadyForAdvanceEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForAdvanceEventOutputTerminal object.
 	(ScopeReadyForAdvanceEventOutputTerminal to String)	Converts a ScopeReadyForAdvanceEventOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Conversion (String to ScopeReadyForAdvanceEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForAdvanceEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReadyForAdvanceEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeReadyForAdvanceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForAdvanceEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

Returns the [ScopeReadyForAdvanceEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Conversion (String to ScopeReadyForAdvanceEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForAdvanceEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReadyForAdvanceEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReadyForAdvanceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForAdvanceEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForAdvanceEventOutputTerminal](#)

Returns the [ScopeReadyForAdvanceEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminal Conversion
(ScopeReadyForAdvanceEventOutputTerminal to String)

Converts a [ScopeReadyForAdvanceEventOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (
    ScopeReadyForAdvanceEventOutputTerminal source
)
```

```
Public Shared Widening Operator CType (
    source As ScopeReadyForAdvanceEventOutputTerminal
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the [ScopeReadyForAdvanceEventOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeReadyForAdvanceEventOutputTerminal](#) object.

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)
[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForAdvanceEventOutputTerminalInequality Operator

Checks whether two [ScopeReadyForAdvanceEventOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeReadyForAdvanceEventOutputTerminal source1,
    ScopeReadyForAdvanceEventOutputTerminal source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeReadyForAdvanceEventOutputTerminal,
    source2 As ScopeReadyForAdvanceEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the first [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForAdvanceEventOutputTerminal](#)
Specifies the second [ScopeReadyForAdvanceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeReadyForAdvanceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Class

Represents the output terminal for [ReadyForReferenceEventOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB




















```
public sealed class ScopeReadyForReferenceEventOutputTerminal
```

```
Public NotInheritable Class ScopeReadyForReferenceEventOutputTerminal
```

The `ScopeReadyForReferenceEventOutputTerminal` type exposes the following members.



Properties






	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.

 Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
 Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
 Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
 Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
 Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
 ClockOut	Gets the output terminal for CLK OUT.
 NoSource	Gets the output terminal for no source.
 Pfi0	Gets the output terminal for PFI 0.
 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)









Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeReadyForReferenceEventOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForReferenceEventOutputTerminal)	Determines whether the current instance of <code>ScopeReadyForReferenceEventOutputTerminal</code> and the specified <code>ScopeReadyForReferenceEventOutputTerminal</code> object are equal.

 FromString 	Creates a <code>ScopeReadyForReferenceEventOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeReadyForReferenceEventOutputTerminal</code> . (Overrides <code>Object.GetHashCode</code> .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code> .)
 ToString	Converts the current instance of <code>ScopeReadyForReferenceEventOutputTerminal</code> to <code>String</code> . (Overrides <code>Object.ToString</code> .)

[Top](#)

Operators

Name	Description
 Equality 	Checks whether two <code>ScopeReadyForReferenceEventOutputTerminal</code> object instances are equal.
 (String to ScopeReadyForReferenceEventOutputTerminal) 	Converts a specified <code>String</code> to an equivalent <code>ScopeReadyForReferenceEventOutputTerminal</code> object.
 (ScopeReadyForReferenceEventOutputTerminal to String) 	Converts a <code>ScopeReadyForReferenceEventOutputTerminal</code> object to an equivalent <code>String</code> .
 Inequality 	Checks whether two <code>ScopeReadyForReferenceEventOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference












[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsReadyForReferenceEventOutputTerminal](#)












Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Properties

The [ScopeReadyForReferenceEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.

 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi2

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)
 A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalClock

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi0

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)
A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi1

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi2

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPxiStar

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static ScopeReadyForReferenceEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi0

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi1

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi2

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi3

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As  
ScopeReadyForReferenceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi4

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeReadyForReferenceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi5

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi6

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi7

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)























[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Properties

The [ScopeReadyForReferenceEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	<u>Aux0Pfi0</u>	Gets the output terminal for AUX I/O connector 0 PFI 0.
	<u>Aux0Pfi1</u>	Gets the output terminal for AUX I/O connector 0 PFI 1.
	<u>Aux0Pfi2</u>	Gets the output terminal for AUX I/O connector 0 PFI 2.
	<u>Aux0Pfi3</u>	Gets the output terminal for AUX I/O connector 0 PFI 3.
	<u>Aux0Pfi4</u>	Gets the output terminal for AUX I/O connector 0 PFI 4.
	<u>Aux0Pfi5</u>	Gets the output terminal for AUX I/O connector 0 PFI 5.
	<u>Aux0Pfi6</u>	Gets the output terminal for AUX I/O connector 0 PFI 6.
	<u>Aux0Pfi7</u>	Gets the output terminal for AUX I/O connector 0 PFI 7.
	<u>ClockOut</u>	Gets the output terminal for CLK OUT.
	<u>NoSource</u>	Gets the output terminal for no source.
	<u>Pfi0</u>	Gets the output terminal for PFI 0.
	<u>Pfi1</u>	Gets the output terminal for PFI 1.
	<u>Pfi2</u>	Gets the output terminal for PFI 2.
	<u>PxiStar</u>	Gets the output terminal for PXI STAR.
	<u>Rtsi0</u>	Gets the output terminal for RTSI 0.
	<u>Rtsi1</u>	Gets the output terminal for RTSI 1.
	<u>Rtsi2</u>	Gets the output terminal for RTSI 2.
	<u>Rtsi3</u>	Gets the output terminal for RTSI 3.
	<u>Rtsi4</u>	Gets the output terminal for RTSI 4.
	<u>Rtsi5</u>	Gets the output terminal for RTSI 5.
	<u>Rtsi6</u>	Gets the output terminal for RTSI 6.
	<u>Rtsi7</u>	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeReadyForReferenceEventOutputTerminal
    Get
```


Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
    ScopeReadyForReferenceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
    ScopeReadyForReferenceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeReadyForReferenceEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RT5I_5".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As
ScopeReadyForReferenceEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
    ScopeReadyForReferenceEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

A [ScopeReadyForReferenceEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Methods

The [ScopeReadyForReferenceEventOutputTerminal](#) type exposes the following members.

Methods



Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForReferenceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified ScopeReadyForReferenceEventOutputTerminal object are equal.
 FromString 	Creates a ScopeReadyForReferenceEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeReadyForReferenceEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeReadyForReferenceEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)[See Also](#)[Reference](#)[ScopeReadyForReferenceEventOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEqual

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForReferenceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified ScopeReadyForReferenceEventOutputTerminal object are equal.

[Top](#)[See Also](#)[Reference](#)[ScopeReadyForReferenceEventOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEqual

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquals

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified [ScopeReadyForReferenceEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReadyForReferenceEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReadyForReferenceEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the [ScopeReadyForReferenceEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForReferenceEventOutputTerminal)	Determines whether the current instance of ScopeReadyForReferenceEventOutputTerminal and the specified ScopeReadyForReferenceEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEqual

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquals Method (ScopeReadyForReferenceEventOutputTerminal)

Determines whether the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) and the specified [ScopeReadyForReferenceEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReadyForReferenceEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReadyForReferenceEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the [ScopeReadyForReferenceEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalFromString Method

Creates a [ScopeReadyForReferenceEventOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForReferenceEventOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeReadyForReferenceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeReadyForReferenceEventOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

Returns an object of type [ScopeReadyForReferenceEventOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal.GetHashCode Method

Gets the hash code for the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalToString Method

Converts the current instance of [ScopeReadyForReferenceEventOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeReadyForReferenceEventOutputTerminal](#).

See Also

Reference





[ScopeReadyForReferenceEventOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Operators and Type Conversions

The [ScopeReadyForReferenceEventOutputTerminal](#) type exposes the following members.

Operators

Name	Description
 Equality	Checks whether two ScopeReadyForReferenceEventOutputTerminal object instances are equal.
 (String to ScopeReadyForReferenceEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForReferenceEventOutputTerminal object.
 (ScopeReadyForReferenceEventOutputTerminal to String)	Converts a ScopeReadyForReferenceEventOutputTerminal object to an equivalent String .
 Inequality	Checks whether two ScopeReadyForReferenceEventOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEqual

Checks whether two [ScopeReadyForReferenceEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReadyForReferenceEventOutputTerminal source1,
    ScopeReadyForReferenceEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReadyForReferenceEventOutputTerminal,
    source2 As ScopeReadyForReferenceEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the first [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the second [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalEquality Operator

Checks whether two [ScopeReadyForReferenceEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReadyForReferenceEventOutputTerminal source1,
    ScopeReadyForReferenceEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReadyForReferenceEventOutputTerminal,
    source2 As ScopeReadyForReferenceEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the first [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the second [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Conversion Operators

Overload List

	Name	Description
	(String to ScopeReadyForReferenceEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForReferenceEventOutputTerminal object.
	(ScopeReadyForReferenceEventOutputTerminal to String)	Converts a ScopeReadyForReferenceEventOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Conversion (String to ScopeReadyForReferenceEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForReferenceEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReadyForReferenceEventOutputTerminal (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeReadyForReferenceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForReferenceEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

Returns the [ScopeReadyForReferenceEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Conversion (String to ScopeReadyForReferenceEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForReferenceEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReadyForReferenceEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReadyForReferenceEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForReferenceEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForReferenceEventOutputTerminal](#)

Returns the [ScopeReadyForReferenceEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminal Conversion
(ScopeReadyForReferenceEventOutputTerminal to String)

Converts a [ScopeReadyForReferenceEventOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public static implicit operator string (  
    ScopeReadyForReferenceEventOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeReadyForReferenceEventOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
Specifies the [ScopeReadyForReferenceEventOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeReadyForReferenceEventOutputTerminal](#) object.

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForReferenceEventOutputTerminalInequality Operator

Checks whether two [ScopeReadyForReferenceEventOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static bool operator !=(  
    ScopeReadyForReferenceEventOutputTerminal source1,  
    ScopeReadyForReferenceEventOutputTerminal source2  
)
```

```
Public Shared Operator <> (
    source1 As ScopeReadyForReferenceEventOutputTerminal,
    source2 As ScopeReadyForReferenceEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
 Specifies the first [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForReferenceEventOutputTerminal](#)
 Specifies the second [ScopeReadyForReferenceEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeReadyForReferenceEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Class

Represents the output terminal for [ReadyForStartEventOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


















C#VB






```
public sealed class ScopeReadyForStartEventOutputTerminal
```

```
Public NotInheritable Class ScopeReadyForStartEventOutputTerminal
```

The `ScopeReadyForStartEventOutputTerminal` type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.

 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.





[Top](#)

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeReadyForStartEventOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForStartEventOutputTerminal)	Determines whether the current instance of <code>ScopeReadyForStartEventOutputTerminal</code> and the specified <code>ScopeReadyForStartEventOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeReadyForStartEventOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeReadyForStartEventOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeReadyForStartEventOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether two <code>ScopeReadyForStartEventOutputTerminal</code> object instances are equal.
	(String to <code>ScopeReadyForStartEventOutputTerminal</code>)	Converts a specified String to an equivalent <code>ScopeReadyForStartEventOutputTerminal</code> object.
	(<code>ScopeReadyForStartEventOutputTerminal</code> to String)	Converts a <code>ScopeReadyForStartEventOutputTerminal</code> object to an equivalent String .
	Inequality	Checks whether two <code>ScopeReadyForStartEventOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference























[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsReadyForStartEventOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Properties

The [ScopeReadyForStartEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi2

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeReadyForStartEventOutputTerminal
    Get
```


Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi4 F

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi5

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalClockOut

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtSi0 Prop

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi1 Prop

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi2 Prop

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi3 Prop

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi4 Prop

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi5 Prop

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi6 Prop

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi7 Prop

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)























[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Properties

The [ScopeReadyForStartEventOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReadyForStartEventOutputTerminal  
Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public static ScopeReadyForStartEventOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

ScopeReadyForStartEventOutputTerminal Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi4 { get; }
```



```
Public Shared ReadOnly Property Rtsi4 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeReadyForStartEventOutputTerminal  
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeReadyForStartEventOutputTerminal
    Get
```

Property Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

A [ScopeReadyForStartEventOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)








Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Methods

The [ScopeReadyForStartEventOutputTerminal](#) type exposes the following members.

Methods

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForStartEventOutputTerminal)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified ScopeReadyForStartEventOutputTerminal object are equal.
 FromString 	Creates a ScopeReadyForStartEventOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeReadyForStartEventOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeReadyForStartEventOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)

See Also



Reference

[ScopeReadyForStartEventOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Met

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForStartEventOutputTerminal)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified ScopeReadyForStartEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method

Determines whether the current instance of [ScopeReadyForStartEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)
[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method

Determines whether the current instance of [ScopeReadyForStartEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Met

Determines whether the current instance of

[ScopeReadyForStartEventOutputTerminal](#) and the specified

[ScopeReadyForStartEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReadyForStartEventOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReadyForStartEventOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
Specifies the [ScopeReadyForStartEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReadyForStartEventOutputTerminal)	Determines whether the current instance of ScopeReadyForStartEventOutputTerminal and the specified ScopeReadyForStartEventOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method

Determines whether the current instance of [ScopeReadyForStartEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeReadyForStartEventOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquals Method
(ScopeReadyForStartEventOutputTerminal)

Determines whether the current instance of [ScopeReadyForStartEventOutputTerminal](#) and the specified [ScopeReadyForStartEventOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(
    ScopeReadyForStartEventOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeReadyForStartEventOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)

Specifies the [ScopeReadyForStartEventOutputTerminal](#) object to compare to the current instance of [ScopeReadyForStartEventOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalFromString Method

Creates a [ScopeReadyForStartEventOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReadyForStartEventOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeReadyForStartEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeReadyForStartEventOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

Returns an object of type [ScopeReadyForStartEventOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalGetHashCode Method

Gets the hash code for the current instance of ScopeReadyForStartEventOutputTerminal.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: Int32

Returns an Int32 representing the hash code for the current instance of ScopeReadyForStartEventOutputTerminal.

See Also

Reference

ScopeReadyForStartEventOutputTerminal Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalToString Method

Converts the current instance of ScopeReadyForStartEventOutputTerminal to String.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeReadyForStartEventOutputTerminal](#).

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Operators and Type Conversions

The [ScopeReadyForStartEventOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
  	Equality	Checks whether two ScopeReadyForStartEventOutputTerminal object instances are equal.

	(String to ScopeReadyForStartEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForStartEventOutputTerminal object.
	(ScopeReadyForStartEventOutputTerminal to String)	Converts a ScopeReadyForStartEventOutputTerminal object to an equivalent String .
	Inequality	Checks whether two ScopeReadyForStartEventOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquality Op

Checks whether two [ScopeReadyForStartEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static bool operator ==(
    ScopeReadyForStartEventOutputTerminal source1,
    ScopeReadyForStartEventOutputTerminal source2
)
```



```
Public Shared Operator = (
    source1 As ScopeReadyForStartEventOutputTerminal,
    source2 As ScopeReadyForStartEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
 Specifies the first [ScopeReadyForStartEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
 Specifies the second [ScopeReadyForStartEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalEquality Operator

Checks whether two [ScopeReadyForStartEventOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReadyForStartEventOutputTerminal source1,
    ScopeReadyForStartEventOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReadyForStartEventOutputTerminal,
    source2 As ScopeReadyForStartEventOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
Specifies the first [ScopeReadyForStartEventOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
Specifies the second [ScopeReadyForStartEventOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference





[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Conversion Operators

Overload List

	Name	Description
 	(String to ScopeReadyForStartEventOutputTerminal)	Converts a specified String to an equivalent ScopeReadyForStartEventOutputTerminal object.
 	(ScopeReadyForStartEventOutputTerminal to String)	Converts a ScopeReadyForStartEventOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Conversion (String to ScopeReadyForStartEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForStartEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static implicit operator ScopeReadyForStartEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReadyForStartEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForStartEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

Returns the [ScopeReadyForStartEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Conversion (String to ScopeReadyForStartEventOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReadyForStartEventOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReadyForStartEventOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReadyForStartEventOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReadyForStartEventOutputTerminal](#) object.

Return Value

Type: [ScopeReadyForStartEventOutputTerminal](#)

Returns the [ScopeReadyForStartEventOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminal Conversion (ScopeReadyForStartEventOutputTerminal to String)

Converts a [ScopeReadyForStartEventOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeReadyForStartEventOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeReadyForStartEventOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal](#)
Specifies the [ScopeReadyForStartEventOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeReadyForStartEventOutputTerminal](#) object.

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)
[Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReadyForStartEventOutputTerminalInequality Operator

Checks whether two ScopeReadyForStartEventOutputTerminal object instances are not equal.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeReadyForStartEventOutputTerminal source1,  
    ScopeReadyForStartEventOutputTerminal source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeReadyForStartEventOutputTerminal,  
    source2 As ScopeReadyForStartEventOutputTerminal  
) As Boolean
```

Parameters

source1

Type: NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal
Specifies the first ScopeReadyForStartEventOutputTerminal object to compare.

source2

Type: NationalInstruments.ModularInstruments.NIScopeScopeReadyForStartEventOutputTerminal
Specifies the second ScopeReadyForStartEventOutputTerminal object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeReadyForStartEventOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevel Class

Provides properties and methods for configuring reference levels used in scalar measurements.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeReferenceLevel](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax




C#VB

```
public sealed class ScopeReferenceLevel : ScopeSubObject,
    IIVI_ScopeReferenceLevel
```

```
Public NotInheritable Class ScopeReferenceLevel
    Inherits ScopeSubObject
    Implements IIVI_ScopeReferenceLevel
```





The `ScopeReferenceLevel` type exposes the following members.

Properties

	Name	Description
	High	Gets or sets the high reference level used in scalar measurements.
	Low	Gets or sets the low reference level used in scalar measurements.
	Mid	Gets or sets the mid reference level used in scalar measurements.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Reference Levels topic in the [NI High-Speed Digitizers Help](#) or the Attributes section in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevel Properties

The [ScopeReferenceLevel](#) type exposes the following members.

Properties

	Name	Description
	High	Gets or sets the high reference level used in scalar measurements.
	Low	Gets or sets the low reference level used in scalar measurements.
	Mid	Gets or sets the mid reference level used in scalar measurements.

[Top](#)

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelHigh Property

Gets or sets the high reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The high reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelHigh](#)

Exceptions

Exception	Condition
ObjectDisposedException	High was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelHigh Property

Gets or sets the high reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The high reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelHigh](#)

Exceptions

Exception	Condition
ObjectDisposedException	High was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelLow Property

Gets or sets the low reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public double Low { get; set; }
```

```
Public Property Low As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The low reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelLow](#)

Exceptions

Exception	Condition
ObjectDisposedException	Low was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelMid Property

Gets or sets the mid reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Mid { get; set; }
```

```
Public Property Mid As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The mid reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelMid](#)

Exceptions

Exception	Condition
ObjectDisposedException	Mid was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevel Properties

The [ScopeReferenceLevel](#) type exposes the following members.

Properties

	Name	Description
	High	Gets or sets the high reference level used in scalar measurements.
	Low	Gets or sets the low reference level used in scalar measurements.
	Mid	Gets or sets the mid reference level used in scalar measurements.

[Top](#)

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelHigh Property

Gets or sets the high reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The high reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelHigh](#)

Exceptions

Exception	Condition
ObjectDisposedException	High was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelHigh Property

Gets or sets the high reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double High { get; set; }
```

```
Public Property High As Double
    Get
    Set
```


Property Value

Type: [Double](#)

The high reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelHigh](#)

Exceptions

Exception	Condition
ObjectDisposedException	High was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelLow Property

Gets or sets the low reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public double Low { get; set; }
```

```
Public Property Low As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The low reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelLow](#)

Exceptions

Exception	Condition
ObjectDisposedException	Low was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevelMid Property

Gets or sets the mid reference level used in scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public double Mid { get; set; }
```

```
Public Property Mid As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The mid reference level used in scalar measurements.

Implements

[IviScopeReferenceLevelMid](#)

Exceptions

Exception	Condition
ObjectDisposedException	Mid was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

See Also

Reference

[ScopeReferenceLevel Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceLevel Methods

The [ScopeReferenceLevel](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object.

		(Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeReferenceLevel Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTrigger Class

Represents the [ScopeMeasurement](#) sub-object that contains properties and methods that are related to reference triggers.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeReferenceTrigger](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax







[C#](#)[VB](#)

```
public sealed class ScopeReferenceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeReferenceTrigger
    Inherits ScopeSubObject
```




The `ScopeReferenceTrigger` type exposes the following members.



Properties

Name	Description
 Delay	Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.
 DetectorLocation	Gets or sets which reference trigger detection circuitry to use on the instrument.
 Hysteresis	Specifies the size of the hysteresis window on either side of the trigger level.
 QuietTimeMin	Gets or sets the amount of time the trigger circuit must not detect a signal above the trigger level, or below the trigger level if the trigger slope is negative, before the trigger is armed.
 ReferencePosition	Gets or sets the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time specified by Delay . The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.
 Window	Gets the ScopeWindowReferenceTrigger sub-object.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)

 SendSoftwareEdgeTrigger	Sends the Reference Trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Configure Trigger topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTrigger Properties

The [ScopeReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 Delay	Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

 DetectorLocation	Gets or sets which reference trigger detection circuitry to use on the instrument.
 Hysteresis	Specifies the size of the hysteresis window on either side of the trigger level.
 QuietTimeMin	Gets or sets the amount of time the trigger circuit must not detect a signal above the trigger level, or below the trigger level if the trigger slope is negative, before the trigger is armed.
 ReferencePosition	Gets or sets the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time specified by Delay . The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.
 Window	Gets the ScopeWindowReferenceTrigger sub-object.

[Top](#)

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDelay Property

Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public PrecisionTimeSpan Delay { get; set; }
```

```
Public Property Delay As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDelay Property

Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)


```
public PrecisionTimeSpan Delay { get; set; }
```

```
Public Property Delay As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDetectorLocation Property

Gets or sets which reference trigger detection circuitry to use on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTriggerDetectorLocation DetectorLocation { get; set; }
```

```
Public Property DetectorLocation As ScopeReferenceTriggerDetectorLocation
    Get
    Set
```

Property Value

Type: [ScopeReferenceTriggerDetectorLocation](#)

Specifies the [ScopeReferenceTriggerDetectorLocation](#). The default value is [AnalogDetectionCircuit](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerDetectorLocation](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerHysteresis Property

Specifies the size of the hysteresis window on either side of the trigger level.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

[C#VB](#)

```
public double Hysteresis { get; set; }
```

```
Public Property Hysteresis As Double
    Get
    Set
```

Property Value

Type: [Double](#)

Units: Volts

Min Value: 0

Max Value for positive trigger slope: $\text{Hysteresis} - \text{trigger level} \geq -(\text{vertical range}/2) + \text{vertical offset}$

Max Value for negative trigger slope: $\text{Hysteresis} + \text{trigger level} \leq -(\text{vertical range}/2) + \text{vertical offset}$

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerQuietTimeMin Property

Gets or sets the amount of time the trigger circuit must not detect a signal above the trigger level, or below the trigger level if the trigger slope is negative, before the trigger is armed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan QuietTimeMin { get; set; }
```

```
Public Property QuietTimeMin As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Any value greater than equal to 0.

Remarks

Before setting this property to any positive value, complete the following steps:

1. Set the trigger [Type](#) to [Edge](#) trigger type.
2. Set the DDC [Enabled](#) to true on channel 0.
3. Set the [DdcDataProcessingMode](#) to [Complex](#)
4. Set the reference trigger [DetectorLocation](#) to [DdcOutput](#).
5. Set the reference trigger [Source](#) to channel "0."

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.PrecisionTimeSpan](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerReferencePosition Property

Gets or sets the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time specified by [Delay](#). The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ReferencePosition { get; set; }
```

```
Public Property ReferencePosition As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The valid values for this property range from 0.0 to 100.0.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerWindow Property

Gets the [ScopeWindowReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWindowReferenceTrigger Window { get; }
```

```
Public ReadOnly Property Window As ScopeWindowReferenceTrigger  
    Get
```

Property Value

Type: [ScopeWindowReferenceTrigger](#)

An object of type [ScopeWindowReferenceTrigger](#).

See Also

Reference

[ScopeReferenceTrigger Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTrigger Properties

The [ScopeReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 Delay	Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.
 DetectorLocation	Gets or sets which reference trigger detection circuitry to use on the instrument.
 Hysteresis	Specifies the size of the hysteresis window on either side of the trigger level.
 QuietTimeMin	Gets or sets the amount of time the trigger circuit must not detect a signal above the trigger level, or below the trigger level if the trigger slope is negative, before the trigger is armed.
 ReferencePosition	Gets or sets the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time specified by Delay . The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.
 Window	Gets the ScopeWindowReferenceTrigger sub-object.

[Top](#)

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDelay Property

Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan Delay { get; set; }
```

```
Public Property Delay As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDelay Property

Gets or sets the trigger delay time, in seconds. The trigger delay time is the length of time the digitizer waits after receiving the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan Delay { get; set; }
```

```
Public Property Delay As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDetectorLocation Property

Gets or sets which reference trigger detection circuitry to use on the instrument.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTriggerDetectorLocation DetectorLocation { get; set; }
```

```
Public Property DetectorLocation As ScopeReferenceTriggerDetectorLocation
    Get
    Set
```

Property Value

Type: [ScopeReferenceTriggerDetectorLocation](#)

Specifies the [ScopeReferenceTriggerDetectorLocation](#). The default value is [AnalogDetectionCircuit](#).

Remarks

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerDetectorLocation](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerHysteresis Property

Specifies the size of the hysteresis window on either side of the trigger level.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in

`NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public double Hysteresis { get; set; }
```

```
Public Property Hysteresis As Double
    Get
    Set
```

Property Value

Type: [Double](#)

Units: Volts

Min Value: 0

Max Value for positive trigger slope: $\text{Hysteresis} - \text{trigger level} \geq -(\text{vertical range}/2) + \text{vertical offset}$

Max Value for negative trigger slope: $\text{Hysteresis} + \text{trigger level} \leq -(\text{vertical range}/2) + \text{vertical offset}$

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerQuietTimeMin Property

Gets or sets the amount of time the trigger circuit must not detect a signal above the trigger level, or below the trigger level if the trigger slope is negative, before the trigger is armed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan QuietTimeMin { get; set; }
```

```
Public Property QuietTimeMin As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Any value greater than equal to 0.

Remarks

Before setting this property to any positive value, complete the following steps:

1. Set the trigger [Type](#) to [Edge](#) trigger type.
2. Set the DDC [Enabled](#) to true on channel 0.
3. Set the [DdcDataProcessingMode](#) to [Complex](#)
4. Set the reference trigger [DetectorLocation](#) to [DdcOutput](#).
5. Set the reference trigger [Source](#) to channel "0."

You can access this property only with high-speed digitizers that support onboard signal processing (OSP), such as the NI 5142. If you open a session with an instrument that does not support OSP and you attempt to access this property, a `ModularInstrumentsException` is thrown.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.PrecisionTimeSpan](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerReferencePosition Property

Gets or sets the position of the Reference Event in the waveform record. When the digitizer detects a trigger, it waits the length of time specified by [Delay](#). The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ReferencePosition { get; set; }
```

```
Public Property ReferencePosition As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The valid values for this property range from 0.0 to 100.0.

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerWindow Property

Gets the [ScopeWindowReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWindowReferenceTrigger Window { get; }
```

```
Public ReadOnly Property Window As ScopeWindowReferenceTrigger
    Get
```

Property Value

Type: [ScopeWindowReferenceTrigger](#)

An object of type [ScopeWindowReferenceTrigger](#).

See Also

Reference

[ScopeReferenceTrigger Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTrigger Methods

The [ScopeReferenceTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object.

	(Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the Reference Trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeReferenceTrigger Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerSendSoftwareEdgeTrigger Me

Sends the Reference Trigger to the digitizer. Call this method if [Source](#) has been configured to SoftwareTriggerFunction.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerSendSoftwareEdgeTrigger Method

Sends the Reference Trigger to the digitizer. Call this method if [Source](#) has been configured to SoftwareTriggerFunction.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerDetectorLocation Enumeration

Specifies which reference trigger detection circuitry to use on a device that has an onboard signal processing (OSP) block.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeReferenceTriggerDetectorLocation
```

```
Public Enumeration ScopeReferenceTriggerDetectorLocation
```

Members

Member name	Value	Description
AnalogDetectionCircuit	0	The hardware analog circuitry is used to implement the reference trigger. This option detects trigger conditions by analyzing the unprocessed analog signal.
DdcOutput	1	The onboard signal processing logic is used to implement the reference trigger. This option detects trigger conditions by analyzing the processed digital signal.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[ScopeReferenceTriggerDetectorLocation](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Class

Represents the reference trigger output terminal for [ReferenceTriggerOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax












C#VB












```
public sealed class ScopeReferenceTriggerExportedOutputTerminal
```

```
Public NotInheritable Class ScopeReferenceTriggerExportedOutputTerminal
```

The `ScopeReferenceTriggerExportedOutputTerminal` type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.

 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.









[Top](#)

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeReferenceTriggerExportedOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReferenceTriggerExportedOutputTerminal)	Determines whether the current instance of <code>ScopeReferenceTriggerExportedOutputTerminal</code> and the specified <code>ScopeReferenceTriggerExportedOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeReferenceTriggerExportedOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeReferenceTriggerExportedOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeReferenceTriggerExportedOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two <code>ScopeReferenceTriggerExportedOutputTerminal</code> object instances are equal.
 	(String to ScopeReferenceTriggerExportedOutputTerminal)	Converts a specified <code>String</code> to an equivalent <code>ScopeReferenceTriggerExportedOutputTerminal</code> object.
 	(ScopeReferenceTriggerExportedOutputTerminal to String)	Converts a <code>ScopeReferenceTriggerExportedOutputTerminal</code> object to an equivalent <code>String</code> .
 	Inequality	Checks whether two <code>ScopeReferenceTriggerExportedOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference























[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsReferenceTriggerOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Properties

The [ScopeReferenceTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  

  ScopeReferenceTriggerExportedOutputTerminal  

  Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)
 A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#)
 "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalClock

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi0

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi1

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi2

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPxiStar

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi0

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi1

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
ScopeReferenceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi2

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi3

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi4

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_4".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi5

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi6

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi7

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```


Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSL_7".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)















[NationalInstruments.ModularInstruments.NIScope Namespace](#)









Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Properties

The [ScopeReferenceTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.

 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As
    ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As
    ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As  
    ScopeReferenceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As
ScopeReferenceTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As  
    ScopeReferenceTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As  
ScopeReferenceTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

A [ScopeReferenceTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)







Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Methods

The [ScopeReferenceTriggerExportedOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified object are equal.


		(Overrides ObjectEquals(Object).)
	Equals(ScopeReferenceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified ScopeReferenceTriggerExportedOutputTerminal object are equal.
 	FromString	Creates a ScopeReferenceTriggerExportedOutputTerminal object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeReferenceTriggerExportedOutputTerminal . (Overrides ObjectGetHashCode.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Converts the current instance of ScopeReferenceTriggerExportedOutputTerminal to String . (Overrides ObjectToString.)


[Top](#)[See Also](#)[Reference](#)[ScopeReferenceTriggerExportedOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals

Overload List

Name	Description	
	Equals(Object)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object).)

	Equals(ScopeReferenceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified ScopeReferenceTriggerExportedOutputTerminal object are equal.
---	---	--

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```



```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified [ScopeReferenceTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeReferenceTriggerExportedOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeReferenceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
 Specifies the [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals Method

Overload List

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeReferenceTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeReferenceTriggerExportedOutputTerminal and the specified ScopeReferenceTriggerExportedOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeReferenceTriggerExportedOutputTerminalEquals Method](#)
([ScopeReferenceTriggerExportedOutputTerminal](#))

Determines whether the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) and the specified [ScopeReferenceTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeReferenceTriggerExportedOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeReferenceTriggerExportedOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalFromString Method

Creates a [ScopeReferenceTriggerExportedOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeReferenceTriggerExportedOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeReferenceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeReferenceTriggerExportedOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

Returns an object of type [ScopeReferenceTriggerExportedOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalGetHashCode Method

Gets the hash code for the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalToString Method

Converts the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeReferenceTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Operators and Type Conversions

The [ScopeReferenceTriggerExportedOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeReferenceTriggerExportedOutputTerminal object instances are equal.

 (String to ScopeReferenceTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeReferenceTriggerExportedOutputTerminal object.
 (ScopeReferenceTriggerExportedOutputTerminal to String)	Converts a ScopeReferenceTriggerExportedOutputTerminal object to an equivalent String .
 Inequality	Checks whether two ScopeReferenceTriggerExportedOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEqual

Checks whether two [ScopeReferenceTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReferenceTriggerExportedOutputTerminal source1,
    ScopeReferenceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReferenceTriggerExportedOutputTerminal,
    source2 As ScopeReferenceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
 Specifies the first [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
 Specifies the second [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalEquality Operator

Checks whether two [ScopeReferenceTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeReferenceTriggerExportedOutputTerminal source1,
    ScopeReferenceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeReferenceTriggerExportedOutputTerminal,
    source2 As ScopeReferenceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the first [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the second [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Conversion Operators

Overload List

Name	Description
 (String to ScopeReferenceTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeReferenceTriggerExportedOutputTerminal object.
 (ScopeReferenceTriggerExportedOutputTerminal to String)	Converts a ScopeReferenceTriggerExportedOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Conversion (String to ScopeReferenceTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReferenceTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static implicit operator ScopeReferenceTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReferenceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReferenceTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

Returns the [ScopeReferenceTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Conversion (String to ScopeReferenceTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeReferenceTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeReferenceTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeReferenceTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeReferenceTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeReferenceTriggerExportedOutputTerminal](#)

Returns the [ScopeReferenceTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminal Conversion (ScopeReferenceTriggerExportedOutputTerminal to String)

Converts a [ScopeReferenceTriggerExportedOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeReferenceTriggerExportedOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeReferenceTriggerExportedOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the [ScopeReferenceTriggerExportedOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeReferenceTriggerExportedOutputTerminal](#) object.

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

Overload

[NationalInstruments.ModularInstruments.NIScope](#) Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeReferenceTriggerExportedOutputTerminalInequality Operator

Checks whether two [ScopeReferenceTriggerExportedOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeReferenceTriggerExportedOutputTerminal source1,
    ScopeReferenceTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeReferenceTriggerExportedOutputTerminal,
    source2 As ScopeReferenceTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the first [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeReferenceTriggerExportedOutputTerminal](#)
Specifies the second [ScopeReferenceTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeReferenceTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResult Structure

Represents the result of the revision query operation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public struct ScopeRevisionQueryResult
```

```
Public Structure ScopeRevisionQueryResult
```






The `ScopeRevisionQueryResult` type exposes the following members.

Properties

	Name	Description
	FirmwareRevision	Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.
	InstrumentRevision	Gets a string that contains additional version information about NI-SCOPE driver.



[Top](#)

Methods

Name	Description
 Equals(Object)	Determines whether this instance of <code>ScopeRevisionQueryResult</code> and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeRevisionQueryResult)	Determines whether the current instance of <code>ScopeRevisionQueryResult</code> and the <code>ScopeRevisionQueryResult</code> object that you specify are equal.
 GetHashCode	Returns the hash code for <code>ScopeRevisionQueryResult</code> . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

Operators

Name	Description
 Equality	Checks whether the two instances of <code>ScopeRevisionQueryResult</code> are equal.
 Inequality	Checks whether the two instances of <code>ScopeRevisionQueryResult</code> are unequal.

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResult Properties

The [ScopeRevisionQueryResult](#) type exposes the following members.

Properties

Name	Description
 FirmwareRevision	Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.
 InstrumentRevision	Gets a string that contains additional version information about NI-SCOPE driver.

[Top](#)

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultFirmwareRevision Property

Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string FirmwareRevision { get; }
```

```
Public ReadOnly Property FirmwareRevision As String
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument firmware revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultFirmwareRevision Property

Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public string FirmwareRevision { get; }
```

```
Public ReadOnly Property FirmwareRevision As String
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument firmware revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultInstrumentRevision Property

Gets a string that contains additional version information about NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public string InstrumentRevision { get; }
```

```
Public ReadOnly Property InstrumentRevision As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument driver software revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResult Properties

The [ScopeRevisionQueryResult](#) type exposes the following members.

Properties

Name	Description
 FirmwareRevision	Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.
 InstrumentRevision	Gets a string that contains additional version information about NI-SCOPE driver.

[Top](#)

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultFirmwareRevision Property

Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FirmwareRevision { get; }
```

```
Public ReadOnly Property FirmwareRevision As String
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument firmware revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultFirmwareRevision Property

Gets a string that contains the firmware revision information for the NI-SCOPE instrument that you are currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string FirmwareRevision { get; }
```



```
Public ReadOnly Property FirmwareRevision As String
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument firmware revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultInstrumentRevision Property

Gets a string that contains additional version information about NI-SCOPE driver.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string InstrumentRevision { get; }
```

```
Public ReadOnly Property InstrumentRevision As String
    Get
```

Property Value

Type: [String](#)

A [String](#) containing the instrument driver software revision numbers.

See Also

Reference

[ScopeRevisionQueryResult Structure](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResult Methods

The [ScopeRevisionQueryResult](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether this instance of ScopeRevisionQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeRevisionQueryResult)	Determines whether the current instance of ScopeRevisionQueryResult and the ScopeRevisionQueryResult object that you specify are equal.
 GetHashCode	Returns the hash code for ScopeRevisionQueryResult . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

See Also

Reference



[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether this instance of ScopeRevisionQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeRevisionQueryResult)	Determines whether the current instance of ScopeRevisionQueryResult and the ScopeRevisionQueryResult object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (Object)

Determines whether this instance of [ScopeRevisionQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeRevisionQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (Object)

Determines whether this instance of [ScopeRevisionQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeRevisionQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (ScopeRevisionQueryResult)

Determines whether the current instance of [ScopeRevisionQueryResult](#) and the [ScopeRevisionQueryResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeRevisionQueryResult result  
)
```

```
Public Function Equals (  
    result As ScopeRevisionQueryResult  
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
The [ScopeRevisionQueryResult](#) object to be compared to the current instance of [ScopeRevisionQueryResult](#) to be compared.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether this instance of ScopeRevisionQueryResult and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeRevisionQueryResult)	Determines whether the current instance of ScopeRevisionQueryResult and the ScopeRevisionQueryResult object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (Object)

Determines whether this instance of [ScopeRevisionQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeRevisionQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (Object)

Determines whether this instance of [ScopeRevisionQueryResult](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeRevisionQueryResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquals Method (ScopeRevisionQueryResult)

Determines whether the current instance of [ScopeRevisionQueryResult](#) and the [ScopeRevisionQueryResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeRevisionQueryResult result  
)
```

```
Public Function Equals (  
    result As ScopeRevisionQueryResult  
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
The [ScopeRevisionQueryResult](#) object to be compared to the current instance of [ScopeRevisionQueryResult](#) to be compared.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultGetHashCode Method

Returns the hash code for [ScopeRevisionQueryResult](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) containing the hash code for [ScopeRevisionQueryResult](#).

See Also

Reference

[ScopeRevisionQueryResult Structure](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResult Operators

The [ScopeRevisionQueryResult](#) type exposes the following members.

Operators

	Name	Description
  S	Equality	Checks whether the two instances of ScopeRevisionQueryResult are equal.
  S	Inequality	Checks whether the two instances of ScopeRevisionQueryResult are unequal.

[Top](#)

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquality Operator

Checks whether the two instances of [ScopeRevisionQueryResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static bool operator ==(
    ScopeRevisionQueryResult result1,
    ScopeRevisionQueryResult result2
)
```

```
Public Shared Operator = (
    result1 As ScopeRevisionQueryResult,
```

```

        result2 As ScopeRevisionQueryResult
    ) As Boolean

```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultEquality Operator

Checks whether the two instances of [ScopeRevisionQueryResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```

public static bool operator ==(
    ScopeRevisionQueryResult result1,

```

```

        ScopeRevisionQueryResult result2
    )

```

```

Public Shared Operator = (
    result1 As ScopeRevisionQueryResult,
    result2 As ScopeRevisionQueryResult
) As Boolean

```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRevisionQueryResultInequality Operator

Checks whether the two instances of [ScopeRevisionQueryResult](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeRevisionQueryResult result1,
    ScopeRevisionQueryResult result2
)
```

```
Public Shared Operator <> (
    result1 As ScopeRevisionQueryResult,
    result2 As ScopeRevisionQueryResult
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeRevisionQueryResult](#)
A [ScopeRevisionQueryResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeRevisionQueryResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRisMethod Enumeration

Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the maximum sampling rate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeRisMethod
```

```
Public Enumeration ScopeRisMethod
```

Members

Member name	Value	Description
ExactNumberOfAverages	1	Acquires the specified number of records for each bin in the RIS acquisition.
MinimumNumberOfAverages	2	Averages each RIS sample of at least minimum number average randomly distributed points. Any extra points taken are also averaged in.
Incomplete	3	Returns the RIS waveform after the specified timeout even if it is incomplete.
LimitedBinWidth	5	Limits the waveforms in the various bins to be within 200 ps of the center of the bin.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTrigger Class

Represents the sub-object that contains methods and properties used to configure common properties for runt triggering.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeRuntTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax






C#VB


```
public sealed class ScopeRuntTrigger : ScopeSubObject,
    IIVI_ScopeTriggerRunt
```

```
Public NotInheritable Class ScopeRuntTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTriggerRunt
```

The ScopeRuntTrigger type exposes the following members.







Properties

Name	Description
 RuntPolarity	Gets or sets the polarity you want the digitizer to use for runt triggering
 ThresholdHigh	Gets or sets the upper voltage threshold you want the digitizer to use for runt triggering.
 ThresholdLow	Gets or sets the lower voltage threshold you want the digitizer to use for runt triggering.
 TimeCondition	Gets or sets the time condition you want the digitizer to use for time-qualified runt triggering.
 TimeHighLimit	Gets or sets the upper time limit you want the digitizer to use for time-qualified runt triggering.

 TimeLowLimit	Gets or sets the lower time limit you want the digitizer to use for time-qualified runt triggering.
--	---

[Top](#)

Methods

Name	Description
 Configure(String, Double, Double, Polarity)	Configures common properties for analog runt triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog runt triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger related topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources







[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTrigger Properties

The [ScopeRuntTrigger](#) type exposes the following members.

Properties

Name	Description
 RuntPolarity	Gets or sets the polarity you want the digitizer to use for runt triggering
 ThresholdHigh	Gets or sets the upper voltage threshold you want the digitizer to use for runt triggering.
 ThresholdLow	Gets or sets the lower voltage threshold you want the digitizer to use for runt triggering.
 TimeCondition	Gets or sets the time condition you want the digitizer to use for time-qualified runt triggering.
 TimeHighLimit	Gets or sets the upper time limit you want the digitizer to use for time-qualified runt triggering.
 TimeLowLimit	Gets or sets the lower time limit you want the digitizer to use for time-qualified runt triggering.

[Top](#)

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerRuntPolarity Property

Gets or sets the polarity you want the digitizer to use for runt triggering

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerPolarity RuntPolarity { get; set; }
```

```
Public Property RuntPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The runt polarity you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	RuntPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for runt polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when the width polarity is positive, negative, or either.

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerRuntPolarity Property

Gets or sets the polarity you want the digitizer to use for runt triggering

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerPolarity RuntPolarity { get; set; }
```

```
Public Property RuntPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The runt polarity you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	RuntPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for runt polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when the width polarity is positive, negative, or either.

See Also

Reference

[ScopeRunTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerThresholdHigh Property

Gets or sets the upper voltage threshold you want the digitizer to use for runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The upper voltage threshold you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper threshold was invalid.

- or -

The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within the lower and upper threshold range, [ThresholdLow](#).

See Also

Reference

[ScopeRunTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerThresholdLow Property

Gets or sets the lower voltage threshold you want the digitizer to use for runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ThresholdLow { get; set; }
```

```
Public Property ThresholdLow As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The lower voltage threshold you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdLow was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within the lower and upper threshold range, [ThresholdHigh](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerTimeCondition Property

Gets or sets the time condition you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeRuntTriggerCondition TimeCondition { get; set; }
```



```
Public Property TimeCondition As ScopeRuntTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeRuntTriggerCondition](#)

The time condition you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for time condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within or outside the [TimeLowLimit](#) and [TimeHighLimit](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerTimeHighLimit Property

Gets or sets the upper time limit you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimeHighLimit { get; set; }
```

```
Public Property TimeHighLimit As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper time limit you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeHighLimit was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper time limit was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger based on the lower and upper time limit, [TimeLowLimit](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerTimeLowLimit Property

Gets or sets the lower time limit you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimeLowLimit { get; set; }
```

```
Public Property TimeLowLimit As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The lower time limit you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeLowLimit was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower time limit was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger based on the lower and upper time limit, [TimeHighLimit](#).

See Also

Reference

[ScopeRuntTrigger Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTrigger Properties

The [ScopeRuntTrigger](#) type exposes the following members.

Properties

Name	Description
 RuntPolarity	Gets or sets the polarity you want the digitizer to use for runt triggering
 ThresholdHigh	Gets or sets the upper voltage threshold you want the digitizer to use for runt triggering.
 ThresholdLow	Gets or sets the lower voltage threshold you want the digitizer to use for runt triggering.
 TimeCondition	Gets or sets the time condition you want the digitizer to use for time-qualified runt triggering.
 TimeHighLimit	Gets or sets the upper time limit you want the digitizer to use for time-qualified runt triggering.
 TimeLowLimit	Gets or sets the lower time limit you want the digitizer to use for time-qualified runt triggering.

[Top](#)

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerRuntPolarity Property

Gets or sets the polarity you want the digitizer to use for runt triggering

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerPolarity RuntPolarity { get; set; }
```

```
Public Property RuntPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The runt polarity you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	RuntPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for runt polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when the width polarity is positive, negative, or either.

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerRuntPolarity Property

Gets or sets the polarity you want the digitizer to use for runt triggering

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public ScopeTriggerPolarity RuntPolarity { get; set; }
```

```
Public Property RuntPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The runt polarity you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	RuntPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for runt polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when the width polarity is positive, negative, or either.

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerThresholdHigh Property

Gets or sets the upper voltage threshold you want the digitizer to use for runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The upper voltage threshold you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for upper threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within the lower and upper threshold range, [ThresholdLow](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerThresholdLow Property

Gets or sets the lower voltage threshold you want the digitizer to use for runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public double ThresholdLow { get; set; }
```

```
Public Property ThresholdLow As Double
    Get
    Set
```


Property Value

Type: [Double](#)

The lower voltage threshold you want the digitizer to use for runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdLow was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within the lower and upper threshold range, [ThresholdHigh](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerTimeCondition Property

Gets or sets the time condition you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeRunTriggerCondition TimeCondition { get; set; }
```

```
Public Property TimeCondition As ScopeRunTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeRunTriggerCondition](#)

The time condition you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for time condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger when it is within or outside the [TimeLowLimit](#) and [TimeHighLimit](#).

See Also

Reference

[ScopeRunTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerTimeHighLimit Property

Gets or sets the upper time limit you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimeHighLimit { get; set; }
```

```
Public Property TimeHighLimit As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper time limit you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeHighLimit was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper time limit was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger based on the lower and upper time limit, [TimeLowLimit](#).

See Also

Reference

[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerTimeLowLimit Property

Gets or sets the lower time limit you want the digitizer to use for time-qualified runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan TimeLowLimit { get; set; }
```

```
Public Property TimeLowLimit As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The lower time limit you want the digitizer to use for time-qualified runt triggering.

Exceptions

Exception	Condition
ObjectDisposedException	TimeLowLimit was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower time limit was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer will trigger based on the lower and upper time limit, [TimeHighLimit](#).

See Also

Reference

[ScopeRuntTrigger Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTrigger Methods

The [ScopeRuntTrigger](#) type exposes the following members.

Methods

Name	Description
 Configure(String, Double, Double, Polarity)	Configures common properties for analog runt triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog runt triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference



[ScopeRuntTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method

Overload List

Name	Description
 Configure(String, Double, Double, Polarity)	Configures common properties for analog runt triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog runt triggering.

[Top](#)

See Also

Reference

[ScopeRunTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method (String, Double, Double, Polarity)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void Configure(
    string triggerSource,
    double triggerLowThreshold,
    double triggerHighThreshold,
    Polarity triggerPolarity
)
```

```
Public Sub Configure (
    triggerSource As String,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As Polarity
)
```

Parameters

triggerSource

Type: [SystemString](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger, must be less than triggerHighThreshold.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger, must be greater than triggerLowThreshold.

triggerPolarity

Type: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

Implements

[IviScopeTriggerRuntConfigure\(String, Double, Double, Polarity\)](#)

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	Configure(String, Double, Double, Polarity) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for runt polarity was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeRuntTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerConfigure Method (String, Double, Double, Polarity)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    string triggerSource,
    double triggerLowThreshold,
    double triggerHighThreshold,
```



```

        Polarity triggerPolarity
    )

```

```

Public Sub Configure (
    triggerSource As String,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As Polarity
)

```

Parameters

triggerSource

Type: [SystemString](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger, must be less than triggerHighThreshold.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger, must be greater than triggerLowThreshold.

triggerPolarity

Type: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

Implements

[IviScopeTriggerRuntConfigure\(String, Double, Double, Polarity\)](#)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, Double, Polarity) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or -

The value for upper threshold was invalid.
- or -
The value for runt polarity was invalid.
- or -
The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeRuntTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerConfigure Method (ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    double triggerLowThreshold,  
    double triggerHighThreshold,  
    ScopeTriggerPolarity triggerPolarity,  
    ScopeTriggerCoupling triggerCoupling,
```

```
PrecisionTimeSpan triggerHoldoff,
PrecisionTimeSpan triggerDelay
)
```

```
Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As ScopeTriggerPolarity,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: NationalInstrumentsPrecisionTimeSpan

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for runt polarity was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeRuntTrigger Class](#)



[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method

Overload List

Name	Description
 Configure(String, Double, Double, Polarity)	Configures common properties for analog runt triggering.
 Configure(ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog runt triggering.

[Top](#)

See Also

Reference

[ScopeRunTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method (String, Double, Double, Polarity)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void Configure(
    string triggerSource,
    double triggerLowThreshold,
    double triggerHighThreshold,
```

```

        Polarity triggerPolarity
    )

```

```

Public Sub Configure (
    triggerSource As String,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As Polarity
)

```

Parameters

triggerSource

Type: [SystemString](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger, must be less than triggerHighThreshold.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger, must be greater than triggerLowThreshold.

triggerPolarity

Type: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

Implements

[IIviScopeTriggerRuntConfigure\(String, Double, Double, Polarity\)](#)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, Double, Polarity) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or -

The value for upper threshold was invalid.
 - or -
 The value for runt polarity was invalid.
 - or -
 The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeRunTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method (String, Double, Double, Polarity)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    string triggerSource,
    double triggerLowThreshold,
    double triggerHighThreshold,
    Polarity triggerPolarity
)
```

```
Public Sub Configure (
    triggerSource As String,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
```

```
triggerPolarity As Polarity
```

```
)
```

Parameters

triggerSource

Type: [SystemString](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger, must be less than `triggerHighThreshold`.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger, must be greater than `triggerLowThreshold`.

triggerPolarity

Type: `Ivi.ScopePolarity`

The polarity that the width will trigger, either positive, negative, or either.

Implements

`IviScopeTriggerRuntConfigure(String, Double, Double, Polarity)`

Exceptions

Exception	Condition
ObjectDisposedException	<code>Configure(String, Double, Double, Polarity)</code> was called after the associated NIScope or ScopeDriverUtility object was disposed.
<code>IviCDriverException</code>	<p>The value for lower threshold was invalid.</p> <p>- or -</p> <p>The value for upper threshold was invalid.</p> <p>- or -</p> <p>The value for runt polarity was invalid.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeRunTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRunTriggerConfigure Method (ScopeTriggerSource, Double, Double, ScopeTriggerPolarity, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog runt triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    double triggerLowThreshold,  
    double triggerHighThreshold,  
    ScopeTriggerPolarity triggerPolarity,  
    ScopeTriggerCoupling triggerCoupling,  
    PrecisionTimeSpan triggerHoldoff,  
    PrecisionTimeSpan triggerDelay  
)
```

```
Public Sub Configure (  
    triggerSource As ScopeTriggerSource,  
    triggerLowThreshold As Double,  
    triggerHighThreshold As Double,  
    triggerPolarity As ScopeTriggerPolarity,  
    triggerCoupling As ScopeTriggerCoupling,  
    triggerHoldoff As PrecisionTimeSpan,
```

```
triggerDelay As PrecisionTimeSpan
```

```
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerLowThreshold

Type: [SystemDouble](#)

A value indicating the lower threshold for the runt to trigger.

triggerHighThreshold

Type: [SystemDouble](#)

A value indicating the higher threshold for the runt to trigger

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure(ScopeTriggerSource , Double, Double, ScopeTriggerPolarity , ScopeTriggerCoupling , PrecisionTimeSpan , PrecisionTimeSpan) was

	called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for runt polarity was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeRuntTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeRuntTriggerCondition Enumeration

Specifies whether a runt that is within or outside the high and low time limit triggers the oscilloscope.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public enum ScopeRunTriggerCondition
```

```
Public Enumeration ScopeRunTriggerCondition
```

Members

Member name	Value	Description
None	0	Do not use time-qualification for runt trigger.
Within	1	Width of runt is within time limits.
Outside	2	Width of runt is outside time limits.

Remarks

The end-user specifies the [TimeLowLimit](#) and [TimeHighLimit](#)

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Class

Represents the output terminal for [SampleClockOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax























[C#](#)/[VB](#)

```
public sealed class ScopeSampleClockExportedOutputTerminal
```

```
Public NotInheritable Class ScopeSampleClockExportedOutputTerminal
```








The `ScopeSampleClockExportedOutputTerminal` type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.







[Top](#)


Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeSampleClockExportedOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeSampleClockExportedOutputTerminal)	Determines whether the current instance of <code>ScopeSampleClockExportedOutputTerminal</code> and the specified <code>ScopeSampleClockExportedOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeSampleClockExportedOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeSampleClockExportedOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeSampleClockExportedOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

Name	Description
 Equality 	Checks whether two <code>ScopeSampleClockExportedOutputTerminal</code> object instances are equal.
 (String to ScopeSampleClockExportedOutputTerminal) 	Converts a specified <code>String</code> to an equivalent <code>ScopeSampleClockExportedOutputTerminal</code> object.
 (ScopeSampleClockExportedOutputTerminal to String) 	Converts a <code>ScopeSampleClockExportedOutputTerminal</code> object to an equivalent <code>String</code> .

	<u>Inequality</u>	Checks whether two <code>ScopeSampleClockExportedOutputTerminal</code> object instances are not equal.
---	-------------------	--

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)




[ScopeExportedSignalsSampleClockOutputTerminal](#)




















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Properties

The [ScopeSampleClockExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.

 Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
 Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
 Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
 Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
 Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
 ClockOut	Gets the output terminal for CLK OUT.
 NoSource	Gets the output terminal for no source.
 Pfi0	Gets the output terminal for PFI 0.
 Pfi1	Gets the output terminal for PFI 1.
 Pfi2	Gets the output terminal for PFI 2.
 PxiStar	Gets the output terminal for PXI STAR.
 Rtsi0	Gets the output terminal for RTSI 0.
 Rtsi1	Gets the output terminal for RTSI 1.
 Rtsi2	Gets the output terminal for RTSI 2.
 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi2

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi5

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalClockOut

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi0 Prop

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi1 Prop

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi2 Prop

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPxiStar P

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi3 Pro

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi4 Pro

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeSampleClockExportedOutputTerminal
    Get
```


Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_7".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)























[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Properties

The [ScopeSampleClockExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As  
ScopeSampleClockExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeSampleClockExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As
    ScopeSampleClockExportedOutputTerminal
    Get
```


Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeSampleClockExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

A [ScopeSampleClockExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Methods

The [ScopeSampleClockExportedOutputTerminal](#) type exposes the following members.

Methods

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeSampleClockExportedOutputTerminal)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified ScopeSampleClockExportedOutputTerminal object are equal.
 FromString 	Creates a ScopeSampleClockExportedOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeSampleClockExportedOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeSampleClockExportedOutputTerminal to String . (Overrides ObjectToString .)



[Top](#)[See Also](#)[Reference](#)[ScopeSampleClockExportedOutputTerminal Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals M

Overload List

Name	Description
------	-------------

 Equals(Object)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeSampleClockExportedOutputTerminal)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified ScopeSampleClockExportedOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals M

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals M

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)
[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals M

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified [ScopeSampleClockExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool Equals(
    ScopeSampleClockExportedOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeSampleClockExportedOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the [ScopeSampleClockExportedOutputTerminal](#) object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)


[Equals Overload](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified object are equal.

	(Overrides ObjectEquals(Object).)
 Equals(ScopeSampleClockExportedOutputTerminal)	Determines whether the current instance of ScopeSampleClockExportedOutputTerminal and the specified ScopeSampleClockExportedOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals M

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquals Method
(ScopeSampleClockExportedOutputTerminal)

Determines whether the current instance of [ScopeSampleClockExportedOutputTerminal](#) and the specified [ScopeSampleClockExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(
    ScopeSampleClockExportedOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeSampleClockExportedOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the [ScopeSampleClockExportedOutputTerminal](#) object to compare to the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalFromString Method

Creates a [ScopeSampleClockExportedOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockExportedOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeSampleClockExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeSampleClockExportedOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

Returns an object of type [ScopeSampleClockExportedOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

[ScopeSampleClockExportedOutputTerminalGetHashCode Method](#)

Gets the hash code for the current instance of [ScopeSampleClockExportedOutputTerminal](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeSampleClockExportedOutputTerminal](#).

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalToString Method

Converts the current instance of [ScopeSampleClockExportedOutputTerminal](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeSampleClockExportedOutputTerminal](#).

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Operators and Type Conversions

The [ScopeSampleClockExportedOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeSampleClockExportedOutputTerminal object instances are equal.
	(String to ScopeSampleClockExportedOutputTerminal)	Converts a specified String to an equivalent ScopeSampleClockExportedOutputTerminal object.

 	(ScopeSampleClockExportedOutputTerminal to String)	Converts a ScopeSampleClockExportedOutputTerminal object to an equivalent String .
 	Inequality	Checks whether two ScopeSampleClockExportedOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquality

Checks whether two [ScopeSampleClockExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeSampleClockExportedOutputTerminal source1,
    ScopeSampleClockExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeSampleClockExportedOutputTerminal,
    source2 As ScopeSampleClockExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the first [ScopeSampleClockExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the second [ScopeSampleClockExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalEquality Operator

Checks whether two [ScopeSampleClockExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeSampleClockExportedOutputTerminal source1,
```

```
ScopeSampleClockExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeSampleClockExportedOutputTerminal,
    source2 As ScopeSampleClockExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the first [ScopeSampleClockExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
 Specifies the second [ScopeSampleClockExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Conversion Operators

Overload List

Name	Description
------	-------------

 	(String to ScopeSampleClockExportedOutputTerminal)	Converts a specified String to an equivalent ScopeSampleClockExportedOutputTerminal object.
 	(ScopeSampleClockExportedOutputTerminal to String)	Converts a ScopeSampleClockExportedOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Conversion (String to ScopeSampleClockExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeSampleClockExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeSampleClockExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeSampleClockExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeSampleClockExportedOutputTerminal](#) object.

Return Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

Returns the [ScopeSampleClockExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Conversion (String to ScopeSampleClockExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeSampleClockExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeSampleClockExportedOutputTerminal (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeSampleClockExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeSampleClockExportedOutputTerminal](#) object.

Return Value

Type: [ScopeSampleClockExportedOutputTerminal](#)

Returns the [ScopeSampleClockExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminal Conversion
(ScopeSampleClockExportedOutputTerminal to String)

Converts a [ScopeSampleClockExportedOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeSampleClockExportedOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeSampleClockExportedOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
Specifies the [ScopeSampleClockExportedOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeSampleClockExportedOutputTerminal](#) object.

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockExportedOutputTerminalInequality Operator

Checks whether two [ScopeSampleClockExportedOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeSampleClockExportedOutputTerminal source1,  
    ScopeSampleClockExportedOutputTerminal source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeSampleClockExportedOutputTerminal,  
    source2 As ScopeSampleClockExportedOutputTerminal  
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
Specifies the first [ScopeSampleClockExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockExportedOutputTerminal](#)
Specifies the second [ScopeSampleClockExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeSampleClockExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Class

Represents the source terminal for [SampleClockTimebaseSource](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax





C#VB




```
public sealed class ScopeSampleClockTimebaseSource
```

```
Public NotInheritable Class ScopeSampleClockTimebaseSource
```

The `ScopeSampleClockTimebaseSource` type exposes the following members.







Properties

	Name	Description
	ClockIn	Gets the source terminal for CLK IN.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.

 S	Pfi2	Gets the source terminal for PFI 2.
 S	PxieDStarA	Gets the source terminal for PXIe DSTAR A.
 S	PxiStar	Gets the source terminal for PXI STAR.




[Top](#)

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of <code>ScopeSampleClockTimebaseSource</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeSampleClockTimebaseSource)	Determines whether the current instance of <code>ScopeSampleClockTimebaseSource</code> and the specified <code>ScopeSampleClockTimebaseSource</code> object are equal.
 S	FromString	Creates a <code>ScopeSampleClockTimebaseSource</code> object from the specified String .
	GetHashCode	Gets the hash code for the current instance of <code>ScopeSampleClockTimebaseSource</code> . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of <code>ScopeSampleClockTimebaseSource</code> to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 S	Equality	Checks whether two <code>ScopeSampleClockTimebaseSource</code> object instances are equal.
 S	(String to ScopeSampleClockTimebaseSource)	Converts a specified String to an equivalent <code>ScopeSampleClockTimebaseSource</code> object.
 S	(ScopeSampleClockTimebaseSource to String)	Converts a <code>ScopeSampleClockTimebaseSource</code> object to an equivalent String .

**Inequality**

Checks whether two `ScopeSampleClockTimebaseSource` object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[ScopeTimingSampleClockTimebaseSource](#)





[NationalInstruments.ModularInstruments.NIScopeScopeTiming](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Properties

The [ScopeSampleClockTimebaseSource](#) type exposes the following members.

Properties

	Name	Description
	ClockIn	Gets the source terminal for CLK IN.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.

 Pfi2	Gets the source terminal for PFI 2.
 PxieDStarA	Gets the source terminal for PXIe DSTAR A.
 PxiStar	Gets the source terminal for PXI STAR.

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePxieDStarA Prop

Gets the source terminal for PXIe DSTAR A.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource PxieDStarA { get; }
```

```
Public Shared ReadOnly Property PxieDStarA As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PXIE_DSTAR_A".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Properties

The [ScopeSampleClockTimebaseSource](#) type exposes the following members.

Properties

	Name	Description
	ClockIn	Gets the source terminal for CLK IN.
	NoSource	Gets the source terminal for no source.
	Pfi0	Gets the source terminal for PFI 0.
	Pfi1	Gets the source terminal for PFI 1.
	Pfi2	Gets the source terminal for PFI 2.
	PxieDStarA	Gets the source terminal for PXIe DSTAR A.
	PxiStar	Gets the source terminal for PXI STAR.

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceClockIn Property

Gets the source terminal for CLK IN.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource ClockIn { get; }
```

```
Public Shared ReadOnly Property ClockIn As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_CLK_IN".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceNoSource Property

Gets the source terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

ScopeSampleClockTimebaseSource Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePxieDStarA Property

Gets the source terminal for PXIe DSTAR A.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource PxieDStarA { get; }
```

```
Public Shared ReadOnly Property PxieDStarA As ScopeSampleClockTimebaseSource
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PXIE_DSTAR_A".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeSampleClockTimebaseSource  
    Get
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

A [ScopeSampleClockTimebaseSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Methods

The [ScopeSampleClockTimebaseSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeSampleClockTimebaseSource)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified ScopeSampleClockTimebaseSource object are equal.
 	FromString	Creates a ScopeSampleClockTimebaseSource object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeSampleClockTimebaseSource . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeSampleClockTimebaseSource to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeSampleClockTimebaseSource)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified ScopeSampleClockTimebaseSource object are equal.

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (Object)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (Object)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (ScopeSampleClockTimebaseSource)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified [ScopeSampleClockTimebaseSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeSampleClockTimebaseSource source  
)
```

```
Public Function Equals (  
    source As ScopeSampleClockTimebaseSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the [ScopeSampleClockTimebaseSource](#) object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeSampleClockTimebaseSource)	Determines whether the current instance of ScopeSampleClockTimebaseSource and the specified ScopeSampleClockTimebaseSource object are equal.

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (Object)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (Object)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquals Method (ScopeSampleClockTimebaseSource)

Determines whether the current instance of [ScopeSampleClockTimebaseSource](#) and the specified [ScopeSampleClockTimebaseSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeSampleClockTimebaseSource source  
)
```

```
Public Function Equals (  
    source As ScopeSampleClockTimebaseSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the [ScopeSampleClockTimebaseSource](#) object to compare to the current instance of [ScopeSampleClockTimebaseSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceFromString Method

Creates a [ScopeSampleClockTimebaseSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeSampleClockTimebaseSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeSampleClockTimebaseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeSampleClockTimebaseSource](#) object returned represents.

Return Value

Type: [ScopeSampleClockTimebaseSource](#)

Returns an object of type [ScopeSampleClockTimebaseSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceGetHashCode Method

Gets the hash code for the current instance of [ScopeSampleClockTimebaseSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeSampleClockTimebaseSource](#).

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceToString Method

Converts the current instance of [ScopeSampleClockTimebaseSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeSampleClockTimebaseSource](#).

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Operators and Type Conversions

The [ScopeSampleClockTimebaseSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeSampleClockTimebaseSource object instances are equal.
	(String to ScopeSampleClockTimebaseSource)	Converts a specified String to an equivalent ScopeSampleClockTimebaseSource object.
	(ScopeSampleClockTimebaseSource to String)	Converts a ScopeSampleClockTimebaseSource object to an equivalent String .
	Inequality	Checks whether two ScopeSampleClockTimebaseSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquality Operator

Checks whether two [ScopeSampleClockTimebaseSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeSampleClockTimebaseSource source1,
    ScopeSampleClockTimebaseSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeSampleClockTimebaseSource,
    source2 As ScopeSampleClockTimebaseSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the first [ScopeSampleClockTimebaseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the second [ScopeSampleClockTimebaseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceEquality Operator

Checks whether two [ScopeSampleClockTimebaseSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeSampleClockTimebaseSource source1,
    ScopeSampleClockTimebaseSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeSampleClockTimebaseSource,
    source2 As ScopeSampleClockTimebaseSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the first [ScopeSampleClockTimebaseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the second [ScopeSampleClockTimebaseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference





[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Conversion Operators

Overload List

	Name	Description
 	(String to ScopeSampleClockTimebaseSource)	Converts a specified String to an equivalent ScopeSampleClockTimebaseSource object.
 	(ScopeSampleClockTimebaseSource to String)	Converts a ScopeSampleClockTimebaseSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Conversion (String to ScopeSampleClockTimebaseSource)

Converts a specified [String](#) to an equivalent [ScopeSampleClockTimebaseSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public static implicit operator ScopeSampleClockTimebaseSource (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeSampleClockTimebaseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeSampleClockTimebaseSource](#) object.

Return Value

Type: [ScopeSampleClockTimebaseSource](#)

Returns the [ScopeSampleClockTimebaseSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Conversion (String to ScopeSampleClockTimebaseSource)

Converts a specified [String](#) to an equivalent [ScopeSampleClockTimebaseSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeSampleClockTimebaseSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeSampleClockTimebaseSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeSampleClockTimebaseSource](#) object.

Return Value

Type: [ScopeSampleClockTimebaseSource](#)

Returns the [ScopeSampleClockTimebaseSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSource Conversion (ScopeSampleClockTimebaseSource to String)

Converts a [ScopeSampleClockTimebaseSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeSampleClockTimebaseSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeSampleClockTimebaseSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the [ScopeSampleClockTimebaseSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeSampleClockTimebaseSource](#) object.

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleClockTimebaseSourceInequality Operator

Checks whether two [ScopeSampleClockTimebaseSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeSampleClockTimebaseSource source1,
    ScopeSampleClockTimebaseSource source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeSampleClockTimebaseSource,
    source2 As ScopeSampleClockTimebaseSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the first [ScopeSampleClockTimebaseSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSampleClockTimebaseSource](#)
Specifies the second [ScopeSampleClockTimebaseSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeSampleClockTimebaseSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSampleMode Enumeration

Specifies the sample mode the digitizer is currently using.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeSampleMode
```

```
Public Enumeration ScopeSampleMode
```

Members

Member name	Value	Description
Realtime	0	The digitizer is using real-time sampling.
EquivalentTime	1	The digitizer is using equivalent-time sampling.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatistics Class

Maintains the measurement value and the statistical results.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeScalarMeasurementStatistics

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax







C#VB

```
public class ScopeScalarMeasurementStatistics
```

```
Public Class ScopeScalarMeasurementStatistics
```







The `ScopeScalarMeasurementStatistics` type exposes the following members.

Properties

Name	Description
 Max	Gets the largest scalar value acquired in the most recent number of measurements.
 Mean	Gets or sets the mean scalar value of the measurement acquired.
 MeasurementResult	Gets the resulting measurement.
 Min	Gets the smallest scalar value acquired.
 StandardDeviation	Gets the standard deviation of the most recent measurements indicated by StatisticsCount .
 StatisticsCount	Gets the number of times FetchScalarMeasurementStatistics(PrecisionTimeSpan, ScopeScalarMeasurementType) has been called.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 MemberwiseClone	Creates a shallow copy of the current Object . (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the [niScope_FetchMeasurementStats](#) topic in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatistics Properties

The [ScopeScalarMeasurementStatistics](#) type exposes the following members.

Properties

Name	Description
 Max	Gets the largest scalar value acquired in the most recent number of measurements.
 Mean	Gets or sets the mean scalar value of the measurement acquired.
 MeasurementResult	Gets the resulting measurement.
 Min	Gets the smallest scalar value acquired.
 StandardDeviation	Gets the standard deviation of the most recent measurements indicated by StatisticsCount .
 StatisticsCount	Gets the number of times FetchScalarMeasurementStatistics(PrecisionTimeSpan, ScopeScalarMeasurementType) has been called.

[Top](#)

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMax Property

Gets the largest scalar value acquired in the most recent number of measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public double Max { get; }
```

```
Public ReadOnly Property Max As Double
    Get
```

Property Value

Type: [Double](#)

Largest scalar value acquired in the most recent number of measurements.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMax Property

Gets the largest scalar value acquired in the most recent number of measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Max { get; }
```

```
Public ReadOnly Property Max As Double
    Get
```

Property Value

Type: [Double](#)

Largest scalar value acquired in the most recent number of measurements.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMean Property

Gets or sets the mean scalar value of the measurement acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public double Mean { get; }
```

```
Public ReadOnly Property Mean As Double  
    Get
```

Property Value

Type: [Double](#)

Mean scalar value of the measurement.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMeasurementRes

Gets the resulting measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double MeasurementResult { get; }
```

```
Public ReadOnly Property MeasurementResult As Double  
    Get
```

Property Value

Type: [Double](#)

Resulting measurement.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMin Property

Gets the smallest scalar value acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Min { get; }
```

```
Public ReadOnly Property Min As Double  
    Get
```

Property Value

Type: [Double](#)

Smallest scalar value acquired.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsStandardDeviation

Gets the standard deviation of the most recent measurements indicated by [StatisticsCount](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double StandardDeviation { get; }
```

```
Public ReadOnly Property StandardDeviation As Double  
    Get
```

Property Value

Type: [Double](#)

Standard deviation of the most recent measurements indicated by statistics count.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsStatisticsCount F

Gets the number of times [FetchScalarMeasurementStatistics\(PrecisionTimeSpan, ScopeScalarMeasurementType\)](#) has been called.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long StatisticsCount { get; }
```

```
Public ReadOnly Property StatisticsCount As Long
    Get
```

Property Value

Type: [Int64](#)

Number of times `FetchScalarMeasurementStatistics` has been called.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatistics Properties

The [ScopeScalarMeasurementStatistics](#) type exposes the following members.

Properties

Name	Description
 Max	Gets the largest scalar value acquired in the most recent number of measurements.
 Mean	Gets or sets the mean scalar value of the measurement acquired.
 MeasurementResult	Gets the resulting measurement.
 Min	Gets the smallest scalar value acquired.
 StandardDeviation	Gets the standard deviation of the most recent measurements indicated by StatisticsCount .
 StatisticsCount	Gets the number of times FetchScalarMeasurementStatistics(PrecisionTimeSpan, ScopeScalarMeasurementType) has been called.

[Top](#)

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMax Property

Gets the largest scalar value acquired in the most recent number of measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Max { get; }
```

```
Public ReadOnly Property Max As Double  
    Get
```

Property Value

Type: [Double](#)

Largest scalar value acquired in the most recent number of measurements.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMax Property

Gets the largest scalar value acquired in the most recent number of measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Max { get; }
```

```
Public ReadOnly Property Max As Double  
    Get
```

Property Value

Type: [Double](#)

Largest scalar value acquired in the most recent number of measurements.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMean Property

Gets or sets the mean scalar value of the measurement acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Mean { get; }
```

```
Public ReadOnly Property Mean As Double  
    Get
```

Property Value

Type: [Double](#)

Mean scalar value of the measurement.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMeasurementResult Property

Gets the resulting measurement.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double MeasurementResult { get; }
```

```
Public ReadOnly Property MeasurementResult As Double  
    Get
```

Property Value

Type: [Double](#)

Resulting measurement.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsMin Property

Gets the smallest scalar value acquired.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public double Min { get; }
```

```
Public ReadOnly Property Min As Double  
    Get
```

Property Value

Type: [Double](#)

Smallest scalar value acquired.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsStandardDeviation Property

Gets the standard deviation of the most recent measurements indicated by [StatisticsCount](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double StandardDeviation { get; }
```

```
Public ReadOnly Property StandardDeviation As Double  
    Get
```

Property Value

Type: [Double](#)

Standard deviation of the most recent measurements indicated by statistics count.

See Also

Reference

ScopeScalarMeasurementStatistics Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatisticsStatisticsCount Property

Gets the number of times [FetchScalarMeasurementStatistics\(PrecisionTimeSpan, ScopeScalarMeasurementType\)](#) has been called.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long StatisticsCount { get; }
```

```
Public ReadOnly Property StatisticsCount As Long
    Get
```

Property Value

Type: [Int64](#)

Number of times `FetchScalarMeasurementStatistics` has been called.

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementStatistics Methods

The [ScopeScalarMeasurementStatistics](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)

 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 MemberwiseClone	Creates a shallow copy of the current Object . (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeScalarMeasurementStatistics Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeScalarMeasurementType Enumeration

Specifies various scalar measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public enum ScopeScalarMeasurementType
```

```
Public Enumeration ScopeScalarMeasurementType
```

Members

Member name	Value	Description
-------------	-------	-------------

RiseTime	0	<p>The time span, in seconds, from when the waveform crosses the low reference level until it crosses the high reference level.</p> <p>The measurement starts at the left edge of the waveform and finds all low reference level crossings until a high reference level crossing. The final low reference level crossing is used in the calculation. The reference levels are specified by Low and High, and their default values are 10% and 90%.</p>
FallTime	1	<p>The time span, in seconds, from when the waveform crosses the high reference level until it crosses the low reference level. The measurement starts at the left edge of the waveform and finds all high reference level crossings until a low reference level crossing. The final high reference level crossing is used in the calculation.</p> <p>The reference levels are specified by Low and High, and their default values are 10% and 90%.</p>
Frequency	2	The value, in hertz, is 1.0 divided by the Period.
Period	3	<p>The time, in seconds, between the first and third mid reference level crosspoints in seconds. A hysteresis window is applied when finding crosspoints. The mid reference level is 50% by default and is set with Mid.</p>
VoltageRms	4	<p>The value, in volts RMS, determined by the following equation:</p> $\text{Voltage RMS} = \sqrt{(\text{waveform}[i]^2) / \text{numPoints}}$
VoltagePeakToPeak	5	The difference, in volts, of VoltageMax and VoltageMin.

VoltageMax	6	The waveform is searched for its maximum point in volts.
VoltageMin	7	The waveform is searched for the minimum point in volts.
VoltageHigh	8	The last histogram method is used where the voltage high result is the voltage, in volts, of the histogram bin with the maximum number of hits above 60% of the waveform's voltage peak-to-peak value. This calculation is useful for ignoring the overshoot and preshoot on square waves.
VoltageLow	9	The last acquisition histogram method is used where the voltage low result, in volts, is the voltage of the histogram bin with the maximum number of hits below 40% of the waveform's voltage peak-to-peak value. This calculation is useful for ignoring the overshoot and preshoot on square waves.
VoltageAverage	10	The value, in volts, determined by the following equation: Voltage Average = waveform[i] / numPoints
WidthNegative	11	The time difference, in seconds, between the first two mid reference level crossings, where the slopes are negative and positive, respectively. A digital hysteresis is used when finding the crosspoints.
WidthPositive	12	The time difference, in seconds, between the first two mid reference level crossings, where the slopes are positive and negative, respectively. A digital hysteresis is used when finding the crosspoints.
DutyCycleNegative	13	The value, in percentage, is the negative width divided by the 100 times period.

DutyCyclePositive	14	The value, in percentage, is WidthPositive divided by 100 times Period.
VoltageAmplitude	15	The difference of the VoltageHigh and the VoltageLow, in volts.
VoltageCycleRms	16	The number of points in a period, in volts RMS, calculated using the equation: $\text{pointsPerPeriod} = \text{int}(\text{period} / \text{dt} + 0.5)$ where dt is the time between two points and int is a function that returns the integer portion of a floating-point number.
VoltageCycleAverage	17	The number of points in a period in volts calculated using the following equation: $\text{pointsPerPeriod} = \text{int}(\text{period} / \text{dt} + 0.5)$ where dt is the time between two points, and int is a function that returns the integer portion of a floating-point number.
Overshoot	18	The measurement is done on the first edge of the waveform. If two edges exist, the algorithm finds the time interval from the first edge until one half the time to the second edge. The local maxima and minima are found in this interval. If only one edge is present in the waveform, the local maximum and minimum are found between the first edge and the end of the waveform. If the first edge is positive sloped, $\text{overshoot} = 100 * (\text{local maximum} - \text{voltage high}) / \text{voltage amplitude}.$ If the first edge is negative sloped, $\text{overshoot} = 100 * (\text{voltage low} - \text{local minimum}) / \text{voltage amplitude}.$
Preshoot	19	The measurement is done on the second edge of the waveform if two edges exist. The algorithm finds the time interval

		<p>from the middle time between the two edges until the second edge. The local maxima and minima are found in this interval. If only one edge is present in the waveform, the local maximum and minimum are found from the start of the waveform to the first edge.</p> <p>If the edge is negative sloped, preshoot = $100 * (\text{local maximum} - \text{voltage high}) / \text{voltage amplitude}$.</p> <p>If the edge is positive sloped, preshoot = $100 * (\text{voltage low} - \text{local minimum}) / \text{voltage amplitude}$.</p>
LowReferenceVolts	1000	The voltage, in volts, corresponding to the low reference level.
MidReferenceVolts	1001	The voltage, in volts, corresponding to the mid reference level.
HighReferenceVolts	1002	The voltage, in volts, corresponding to the high reference level.
Area	1003	Area = VoltageAverage * numPoints * delta time between two points, in volts*seconds.
CycleArea	1004	<p>Cycle area = voltage cycle average * pointsPerPeriod * dt where pointsPerPeriod is defined in the VoltageCycleAverage description.</p> <p>Units: Volt * Seconds</p>
Integral	1005	<p>Numerical integration is done using Simpson's rule.</p> <p>Units: Volt * Seconds</p>
VoltageBase	1006	<p>If the histogram bin corresponding to voltage low has over 5% of the total hits, the voltage low result is returned. Otherwise, the voltage minimum calculation is returned. This allows using the voltage base to get a reasonable answer for either a square wave</p>

		(ignoring the overshoot and preshoot) or a triangle wave (where a histogram fails). Units: Volts
VoltageTop	1007	If the histogram bin corresponding to VoltageHigh has over 5% of the total hits, the voltage high result is returned. Otherwise, the VoltageMax calculation is returned. This allows using the voltage top to get a reasonable answer for either a square wave (ignoring the overshoot and preshoot) or a triangle wave (where a histogram fails).
FftFrequency	1008	<p>The FFT amplitude spectrum is calculated using a split-radix real FFT, and the frequency, in hertz, corresponding to the maximum amplitude is returned.</p> <p>The FFT amplitude spectrum is calculated using a split-radix real FFT, and the frequency corresponding to the maximum amplitude is returned. If the input waveform size is not a power of two, the waveform is zero-padded to the next higher power of two. The frequency resolution is sampling rate / number of points.</p> <p>The DC bin of the FFT is ignored when searching for the maximum amplitude, so the FFT frequency should ignore any DC offsets. However, the zero-padding used in the FFT measurement can introduce other low-frequency components if the waveform has a large DC offset. To avoid problems, make sure the RecordLength is a power of 2, so no zero padding occurs. You can fetch this property using the RecordLength. Alternately, you can use AC coupling on the digitizer to solve the problem.</p>

FftAmplitude	1009	The FFT amplitude spectrum is calculated using a split-radix real FFT, and the maximum amplitude, in volts RMS, is returned. If the input waveform size is not a power of two, the waveform is zero-padded to the next higher power of two.
RiseSlewRate	1010	The HighReferenceVolts minus the LowReferenceVolts is divided by the RiseTime calculation. Units: Volts/second
FallSlewRate	1011	The value is LowReferenceVolts minus the HighReferenceVolts is divided by the fall-time calculation. The result is always negative. Units: Volts/second
ACEstimate	1012	The DC estimate is subtracted from the waveform, and a Hanning window is applied to give a processed waveform. The RMS voltage is calculated with the following equation: $\sqrt{ \left(\sum \text{processed waveform}[i]^2 \right) / \left(\text{numPoints} * \text{enbw} * \text{cg}^2 \right) },$ where the equivalent noise bandwidth (enbw) for the Hanning window is 1.5, and the coherent gain (cg) is 0.5. This algorithm minimizes the effect of an uneven number of waveform cycles in the measurement, which could arbitrarily increase or decrease the RMS value. Units: Volts RMS
DCEstimate	1013	A Hanning window is applied to give a processed waveform and the voltage average is calculated with the following equation: voltage average = (processed waveform[i]) / (cg * numPoints)

		where the coherent gain (cg) of the Hanning window is 0.5—the DC gain of the window. Units: Volts RMS
TimeDelay	1014	The algorithm finds the first time that the waveform from the channel specified by the channel parameter crosses its mid reference level. Next, the algorithm finds the first two times that the waveform from the channel OtherChannel crosses its mid reference level. The time delay is the time between two edges with the same slope. The mid reference level is stored on a per channel basis, and mid reference levels do not need to be the same. All reference levels use a digital hysteresis. Units: seconds
AveragePeriod	1015	Up to 256 mid reference level crossings are found on the waveform using a digital hysteresis. The time difference between the last crossing and the first crossing is divided by the number of periods found in the waveform. The last crossing is defined as the last crossing in the waveform with the same slope as the first crossing, so an integer number of periods exist in the waveform.
AverageFrequency	1016	The value is 1.0 divided by the AveragePeriod. Units: Hertz
VoltageBaseToTop	1017	The value ofVoltageTop minus VoltageBase.
PhaseDelay	1018	The time delay divided by the period, of the waveform on the channel specified by the measurement method, times 360 degrees. Units: Degrees

TimeHistogramMean	3000	The histogram mean = [(bin hits * bin value)] / total hits. The bin value is the center voltage value of the histogram bin.
TimeHistogramStandardDeviation	3001	Refer to the Time Hist Stdev topic in the NI-SCOPE Function Reference Help .
TimeHistogramPeakToPeak	3002	The difference, in volts, of the Histogram maximum from the the histogram minimum.
TimeHistogramMedian	3003	The bin value, in seconds, where half the histogram hits are above it and half the histogram hits are below.
TimeHistogramHits	3004	The number of points in the histogram.
TimeHistogramMax	3005	The highest bin value with at least one hit.
TimeHistogramMin	3006	The lowest bin value, in seconds, with at least one hit.
TimeHistogramMeanPlusStandardDeviation	3007	The percentage of hits in the histogram between mean minus the standard deviation and mean plus the standard deviation. The percentage is returned in the range 0–100.
TimeHistogramMeanPlus2StandardDeviation	3008	The percentage of hits in the histogram between the mean minus two times the standard deviation and the mean plus two times the standard deviation. The percentage is returned in the range 0–100.
TimeHistogramMeanPlus3StandardDeviation	3009	The percentage of hits in the histogram between the mean minus three times the standard deviation and the mean plus three times the standard deviation. The percentage is returned in the range 0–100.
TimeHistogramMode	3010	The bin value, in seconds, with the most hits. If there is a tie, the lower voltage or time value is returned.

TimeHistogramNewHits	3011	The number of points added to the histogram by the most recent acquisition.
VoltageHistogramMean	2000	The bin value is the center voltage value of the histogram bin. Histogram Mean = $[(\text{bin hits} * \text{bin value})] / \text{total hits}$.
VoltageHistogramStandardDeviation	2001	Refer to the Volt Hist Stdev topic in the NI-SCOPE Function Reference Help .
VoltageHistogramPeakToPeak	2002	The difference of the histogram maximum and the histogram minimum.
VoltageHistogramMedian	2003	The bin value, in volts, where half the histogram hits are above it and half the histogram hits are below.
VoltageHistogramHits	2004	The number of points in the histogram.
VoltageHistogramMax	2005	The highest bin value, in volts, with at least one hit.
VoltageHistogramMin	2006	The lowest bin value, in volts, with at least one hit.
VoltageHistogramMeanPlusStandardDeviation	2007	The percentage of hits in the histogram between mean minus the standard deviation and mean plus the standard deviation. The percentage is returned in the range 0–100.
VoltageHistogramMeanPlus2StandardDeviation	2008	The percentage of hits in the histogram between the mean minus two times the standard deviation and the mean plus two times the standard deviation. The percentage is returned in the range 0–100.
VoltageHistogramMeanPlus3StandardDeviation	2009	The percentage of hits in the histogram between the mean minus three times the standard deviation and the mean plus three times the standard deviation. The percentage is returned in the range 0–100.
VoltageHistogramMode	2010	The bin value, in volts, with the most hits. If there is a tie, the lower voltage or time value is returned.

VoltageHistogramNewHits	2011	The number of points added to the histogram by the most recent acquisition.
-------------------------	------	---

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibration Class

Represents methods related to self calibration.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeSelfCalibration

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB



```
public sealed class ScopeSelfCalibration : ScopeSubObject
```

```
Public NotInheritable Class ScopeSelfCalibration
    Inherits ScopeSubObject
```

The ScopeSelfCalibration type exposes the following members.






Properties

Name	Description
------	-------------

 SelfCalibrationDate	Gets the last self-calibration date.
 SelfCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of self-calibration.

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SelfCalibrate	Self-calibrates the digitizer.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Calibration topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources



[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibration Properties

The [ScopeSelfCalibration](#) type exposes the following members.

Properties

Name	Description
 SelfCalibrationDate	Gets the last self-calibration date.
 SelfCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of self-calibration.

[Top](#)

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationDate Property

Gets the last self-calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public DateTime SelfCalibrationDate { get; }
```

```
Public ReadOnly Property SelfCalibrationDate As DateTime
    Get
```

Property Value

Type: [DateTime](#)

Last self-calibration date.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationDate Property

Gets the last self-calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime SelfCalibrationDate { get; }
```

```
Public ReadOnly Property SelfCalibrationDate As DateTime
    Get
```

Property Value

Type: [DateTime](#)

Last self-calibration date.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationTemperature Prop

Gets the onboard temperature of the digitizer at the time of self-calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SelfCalibrationTemperature { get; }
```

```
Public ReadOnly Property SelfCalibrationTemperature As Double
    Get
```

Property Value

Type: [Double](#)

Onboard temperature of the digitizer at the time of self-calibration.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources



[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibration Properties

The [ScopeSelfCalibration](#) type exposes the following members.

Properties

Name	Description
 SelfCalibrationDate	Gets the last self-calibration date.
 SelfCalibrationTemperature	Gets the onboard temperature of the digitizer at the time of self-calibration.

[Top](#)

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationDate Property

Gets the last self-calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime SelfCalibrationDate { get; }
```

```
Public ReadOnly Property SelfCalibrationDate As DateTime  
    Get
```

Property Value

Type: [DateTime](#)

Last self-calibration date.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationDate Property

Gets the last self-calibration date.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public DateTime SelfCalibrationDate { get; }
```

```
Public ReadOnly Property SelfCalibrationDate As DateTime  
    Get
```

Property Value

Type: [DateTime](#)

Last self-calibration date.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrationTemperature Property

Gets the onboard temperature of the digitizer at the time of self-calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SelfCalibrationTemperature { get; }
```

```
Public ReadOnly Property SelfCalibrationTemperature As Double
    Get
```

Property Value

Type: [Double](#)

Onboard temperature of the digitizer at the time of self-calibration.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources






[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibration Methods

The [ScopeSelfCalibration](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SelfCalibrate	Self-calibrates the digitizer.
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrate Method

Self-calibrates the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SelfCalibrate(  
    ScopeSelfCalibrationOption option  
)
```

```
Public Sub SelfCalibrate (  
    option As ScopeSelfCalibrationOption  
)
```

Parameters

option

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfCalibrationOption](#)
Allows you to self-calibrate all channels or restore the external calibration.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationSelfCalibrate Method

Self-calibrates the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SelfCalibrate(  
    ScopeSelfCalibrationOption option  
)
```

```
Public Sub SelfCalibrate (  
    option As ScopeSelfCalibrationOption  
)
```

Parameters

option

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfCalibrationOption](#)
Allows you to self-calibrate all channels or restore the external calibration.

See Also

Reference

[ScopeSelfCalibration Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfCalibrationOption Enumeration

Specifies how the self-calibration option is carried out.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeSelfCalibrationOption
```

```
Public Enumeration ScopeSelfCalibrationOption
```

Members

Member name	Value	Description
NormalSelfCalibration	0	A normal self-calibration operation.
RestorePreviousExternalCalibration	2	Restores the previous calibration.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResult Structure

Represents the result of [SelfTest](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax



C#VB

```
public struct ScopeSelfTestResult
```

```
Public Structure ScopeSelfTestResult
```






The `ScopeSelfTestResult` type exposes the following members.

Properties

	Name	Description
	Code	Gets the numeric result from the SelfTest operation.
	Message	Gets the string returned from the instrument SelfTest .



Top

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of the <code>ScopeSelfTestResult</code> object and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
	Equals(ScopeSelfTestResult)	Determines whether the current instance of <code>ScopeSelfTestResult</code> and the <code>ScopeSelfTestResult</code> object that you specify are equal.
	GetHashCode	Returns the hash code for <code>ScopeSelfTestResult</code> . (Overrides ValueTypeGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

Top

Operators

	Name	Description
	Equality	Checks whether the two instances of <code>ScopeSelfTestResult</code> are equal.
	Inequality	Checks whether the two instances of <code>ScopeSelfTestResult</code> are unequal.

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResult Properties

The [ScopeSelfTestResult](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the numeric result from the SelfTest operation.
	Message	Gets the string returned from the instrument SelfTest .

[Top](#)

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultCode Property

Gets the numeric result from the [SelfTest](#) operation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) containing the self-test code.

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultCode Property

Gets the numeric result from the [SelfTest](#) operation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) containing the self-test code.

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultMessage Property

Gets the string returned from the instrument [SelfTest](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Message { get; }
```

```
Public ReadOnly Property Message As String
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the string returned from the instrument [SelfTest](#).

See Also

Reference

[ScopeSelfTestResult Structure](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResult Properties

The [ScopeSelfTestResult](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the numeric result from the SelfTest operation.
	Message	Gets the string returned from the instrument SelfTest .

[Top](#)

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultCode Property

Gets the numeric result from the [SelfTest](#) operation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) containing the self-test code.

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultCode Property

Gets the numeric result from the [SelfTest](#) operation.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int Code { get; }
```

```
Public ReadOnly Property Code As Integer  
    Get
```

Property Value

Type: [Int32](#)

An [Int32](#) containing the self-test code.

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultMessage Property

Gets the string returned from the instrument [SelfTest](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Message { get; }
```

```
Public ReadOnly Property Message As String
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the string returned from the instrument [SelfTest](#).

See Also

Reference

[ScopeSelfTestResult Structure](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResult Methods

The [ScopeSelfTestResult](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of the ScopeSelfTestResult object and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeSelfTestResult)	Determines whether the current instance of ScopeSelfTestResult and the ScopeSelfTestResult object that you specify are equal.
 GetHashCode	Returns the hash code for ScopeSelfTestResult . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

See Also

Reference



[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of the ScopeSelfTestResult object and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeSelfTestResult)	Determines whether the current instance of ScopeSelfTestResult and the ScopeSelfTestResult object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (Object)

Determines whether the current instance of the [ScopeSelfTestResult](#) object and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of the [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (Object)

Determines whether the current instance of the [ScopeSelfTestResult](#) object and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of the [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (ScopeSelfTestResult)

Determines whether the current instance of [ScopeSelfTestResult](#) and the [ScopeSelfTestResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeSelfTestResult result  
)
```

```
Public Function Equals (  
    result As ScopeSelfTestResult  
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
The [ScopeSelfTestResult](#) object to be compared to the current instance of [ScopeSelfTestResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of the ScopeSelfTestResult object and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeSelfTestResult)	Determines whether the current instance of ScopeSelfTestResult and the ScopeSelfTestResult object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (Object)

Determines whether the current instance of the [ScopeSelfTestResult](#) object and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of the [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (Object)

Determines whether the current instance of the [ScopeSelfTestResult](#) object and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of the [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquals Method (ScopeSelfTestResult)

Determines whether the current instance of [ScopeSelfTestResult](#) and the [ScopeSelfTestResult](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeSelfTestResult result  
)
```

```
Public Function Equals (  
    result As ScopeSelfTestResult  
) As Boolean
```

Parameters

result

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
The [ScopeSelfTestResult](#) object to be compared to the current instance of [ScopeSelfTestResult](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultGetHashCode Method

Returns the hash code for [ScopeSelfTestResult](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) containing the hash code for [ScopeSelfTestResult](#).

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResult Operators

The [ScopeSelfTestResult](#) type exposes the following members.

Operators

Name	Description
------	-------------

 S	Equality	Checks whether the two instances of ScopeSelfTestResult are equal.
 S	Inequality	Checks whether the two instances of ScopeSelfTestResult are unequal.

[Top](#)

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquality Operator

Checks whether the two instances of [ScopeSelfTestResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static bool operator ==(
    ScopeSelfTestResult result1,
    ScopeSelfTestResult result2
)
```

```
Public Shared Operator = (
    result1 As ScopeSelfTestResult,
    result2 As ScopeSelfTestResult
) As Boolean
```


Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultEquality Operator

Checks whether the two instances of [ScopeSelfTestResult](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeSelfTestResult result1,
    ScopeSelfTestResult result2
)
```

```
Public Shared Operator = (  
    result1 As ScopeSelfTestResult,  
    result2 As ScopeSelfTestResult  
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSelfTestResultInequality Operator

Checks whether the two instances of [ScopeSelfTestResult](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static bool operator !=(
    ScopeSelfTestResult result1,
    ScopeSelfTestResult result2
)
```

```
Public Shared Operator <> (
    result1 As ScopeSelfTestResult,
    result2 As ScopeSelfTestResult
) As Boolean
```

Parameters

result1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

result2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeSelfTestResult](#)
A [ScopeSelfTestResult](#) object.

Return Value

Type: [Boolean](#)

if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeSelfTestResult Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSignalSource Enumeration

Contains values representing signals to use with [ExportSignal\(ScopeSignalSource, String\)](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeSignalSource
```

```
Public Enumeration ScopeSignalSource
```

Members

Member name	Value	Description
None	0	Do not generate a digital pulse.
StartTrigger	2	Generate a pulse when detecting a start trigger.
AdvanceTrigger	5	Generate a pulse when detecting an advance trigger.
ReferenceTrigger	1	Generate a pulse when detecting the stop/reference trigger.
EndOfRecordEvent	4	Generate a pulse at the end of the record.
EndOfAcquisitionEvent	3	Generate a pulse when the acquisition finishes.
ReadyForStartEvent	7	Asserts when the digitizer is initiated and ready to accept a start trigger and begin sampling.
ReadyForAdvanceEvent	6	Asserts when the digitizer is ready to advance to the next record.
ReadyForReferenceEvent	10	Asserts when the digitizer is ready to accept a reference trigger.
ReferenceClock	100	Export the reference clock for the digitizer to the specified terminal.
SampleClock	101	Export the sample clock for the digitizer to the specified terminal.
FiveVoltPower	13	Exports a 5 V power source.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTrigger Class

Represents the ScopeStartTrigger sub-objects that contain properties related to the Start trigger.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeStartTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB

```
public sealed class ScopeStartTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeStartTrigger
    Inherits ScopeSubObject
```

The ScopeStartTrigger type exposes the following members.






Properties

Name	Description
 DigitalEdge	Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

Top

Methods

Name	Description
------	-------------

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the Start Trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction .
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger related topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources


[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTrigger Properties

The [ScopeStartTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

[Top](#)

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerDigitalEdge Property

Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeDigitalEdgeStartTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeStartTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTrigger](#)

The sub-object that contains properties and sub-objects that are related to Digital Edge Start triggers.

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerDigitalEdge Property

Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeStartTrigger  
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTrigger](#)

The sub-object that contains properties and sub-objects that are related to Digital Edge Start triggers.

See Also

Reference

[ScopeStartTrigger Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTrigger Properties

The [ScopeStartTrigger](#) type exposes the following members.

Properties

Name	Description
 DigitalEdge	Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

[Top](#)

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerDigitalEdge Property

Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public ScopeDigitalEdgeStartTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeStartTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTrigger](#)

The sub-object that contains properties and sub-objects that are related to Digital Edge Start triggers.

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerDigitalEdge Property

Gets the parameters for properties and sub-objects that are related to digital edge start triggers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeDigitalEdgeStartTrigger DigitalEdge { get; }
```

```
Public ReadOnly Property DigitalEdge As ScopeDigitalEdgeStartTrigger
    Get
```

Property Value

Type: [ScopeDigitalEdgeStartTrigger](#)

The sub-object that contains properties and sub-objects that are related to Digital Edge Start triggers.

See Also

Reference

[ScopeStartTrigger Class](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTrigger Methods

The [ScopeStartTrigger](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 SendSoftwareEdgeTrigger	Sends the Start Trigger to the digitizer. Call this method if Source has been configured to SoftwareTriggerFunction .
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerSendSoftwareEdgeTrigger Method

Sends the Start Trigger to the digitizer. Call this method if [Source](#) has been configured to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerSendSoftwareEdgeTrigger Method

Sends the Start Trigger to the digitizer. Call this method if [Source](#) has been configured to [SoftwareTriggerFunction](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void SendSoftwareEdgeTrigger()
```

```
Public Sub SendSoftwareEdgeTrigger
```

See Also

Reference

[ScopeStartTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Class

Represents the start trigger exported output terminal for [StartTriggerOutputTerminal](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


















C#VB






```
public sealed class ScopeStartTriggerExportedOutputTerminal
```

```
Public NotInheritable Class ScopeStartTriggerExportedOutputTerminal
```

The `ScopeStartTriggerExportedOutputTerminal` type exposes the following members.








Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.

 Rtsi3	Gets the output terminal for RTSI 3.
 Rtsi4	Gets the output terminal for RTSI 4.
 Rtsi5	Gets the output terminal for RTSI 5.
 Rtsi6	Gets the output terminal for RTSI 6.
 Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)









Methods

Name	Description
 Equals(Object)	Determines whether the current instance of <code>ScopeStartTriggerExportedOutputTerminal</code> and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeStartTriggerExportedOutputTerminal)	Determines whether the current instance of <code>ScopeStartTriggerExportedOutputTerminal</code> and the specified <code>ScopeStartTriggerExportedOutputTerminal</code> object are equal.
 FromString 	Creates a <code>ScopeStartTriggerExportedOutputTerminal</code> object from the specified <code>String</code> .
 GetHashCode	Gets the hash code for the current instance of <code>ScopeStartTriggerExportedOutputTerminal</code> . (Overrides ObjectGetHashCode .)
 GetType	Gets the <code>Type</code> of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of <code>ScopeStartTriggerExportedOutputTerminal</code> to <code>String</code> . (Overrides ObjectToString .)

[Top](#)

Operators

Name	Description
------	-------------

 	Equality	Checks whether two <code>ScopeStartTriggerExportedOutputTerminal</code> object instances are equal.
 	(String to <code>ScopeStartTriggerExportedOutputTerminal</code>)	Converts a specified <code>String</code> to an equivalent <code>ScopeStartTriggerExportedOutputTerminal</code> object.
 	(<code>ScopeStartTriggerExportedOutputTerminal</code> to String)	Converts a <code>ScopeStartTriggerExportedOutputTerminal</code> object to an equivalent <code>String</code> .
 	Inequality	Checks whether two <code>ScopeStartTriggerExportedOutputTerminal</code> object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference























[NationalInstruments.ModularInstruments.NIScope Namespace](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[ScopeExportedSignalsStartTriggerOutputTerminal](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Properties

The [ScopeStartTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi1

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi2

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi3

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi4

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi5

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi6

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi7

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)
 A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#)
 "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalClockOut

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
 NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalNoSource

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPxiStar Pr

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtSi0 Prop

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi1 Prop

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi2 Prop

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi3 Prop

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi4 Prop

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi5 Prop

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi6 Prop

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_6".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)























[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Properties

The [ScopeStartTriggerExportedOutputTerminal](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the output terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the output terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the output terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the output terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the output terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the output terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the output terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the output terminal for AUX I/O connector 0 PFI 7.
	ClockOut	Gets the output terminal for CLK OUT.
	NoSource	Gets the output terminal for no source.
	Pfi0	Gets the output terminal for PFI 0.
	Pfi1	Gets the output terminal for PFI 1.
	Pfi2	Gets the output terminal for PFI 2.
	PxiStar	Gets the output terminal for PXI STAR.
	Rtsi0	Gets the output terminal for RTSI 0.
	Rtsi1	Gets the output terminal for RTSI 1.
	Rtsi2	Gets the output terminal for RTSI 2.
	Rtsi3	Gets the output terminal for RTSI 3.
	Rtsi4	Gets the output terminal for RTSI 4.
	Rtsi5	Gets the output terminal for RTSI 5.
	Rtsi6	Gets the output terminal for RTSI 6.
	Rtsi7	Gets the output terminal for RTSI 7.

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi0

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi0 Property

Gets the output terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi1 Property

Gets the output terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)
 A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi2 Property

Gets the output terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
 Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB


```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi3 Property

Gets the output terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi4 Property

Gets the output terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi5 Property

Gets the output terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi6 Property

Gets the output terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalAux0Pfi7 Property

Gets the output terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalClockOut Property

Gets the output terminal for CLK OUT.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal ClockOut { get; }
```

```
Public Shared ReadOnly Property ClockOut As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_CLK_OUT".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalNoSource Property

Gets the output terminal for no source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal NoSource { get; }
```

```
Public Shared ReadOnly Property NoSource As
ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_NO_SOURCE".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi0 Property

Gets the output terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi1 Property

Gets the output terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPfi2 Property

Gets the output terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalPxiStar Property

Gets the output terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As  
ScopeStartTriggerExportedOutputTerminal  
Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi0 Property

Gets the output terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi1 Property

Gets the output terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi2 Property

Gets the output terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi3 Property

Gets the output terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi4 Property

Gets the output terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi5 Property

Gets the output terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeStartTriggerExportedOutputTerminal
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi6 Property

Gets the output terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalRtsi7 Property

Gets the output terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeStartTriggerExportedOutputTerminal  
    Get
```

Property Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

A [ScopeStartTriggerExportedOutputTerminal](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Methods

The [ScopeStartTriggerExportedOutputTerminal](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeStartTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified ScopeStartTriggerExportedOutputTerminal object are equal.
 FromString 	Creates a ScopeStartTriggerExportedOutputTerminal object from the specified String .
 GetHashCode	Gets the hash code for the current instance of ScopeStartTriggerExportedOutputTerminal . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeStartTriggerExportedOutputTerminal to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Me

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeStartTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified ScopeStartTriggerExportedOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Me

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Me

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Me

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified [ScopeStartTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeStartTriggerExportedOutputTerminal source  
)
```

```
Public Function Equals (  
    source As ScopeStartTriggerExportedOutputTerminal  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the [ScopeStartTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeStartTriggerExportedOutputTerminal)	Determines whether the current instance of ScopeStartTriggerExportedOutputTerminal and the specified ScopeStartTriggerExportedOutputTerminal object are equal.

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Me

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Method (Object)

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquals Method
(ScopeStartTriggerExportedOutputTerminal)

Determines whether the current instance of [ScopeStartTriggerExportedOutputTerminal](#) and the specified [ScopeStartTriggerExportedOutputTerminal](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(
    ScopeStartTriggerExportedOutputTerminal source
)
```

```
Public Function Equals (
    source As ScopeStartTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)

Specifies the [ScopeStartTriggerExportedOutputTerminal](#) object to compare to the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalFromString Method

Creates a [ScopeStartTriggerExportedOutputTerminal](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeStartTriggerExportedOutputTerminal FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeStartTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeStartTriggerExportedOutputTerminal](#) object returned represents.

Return Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

Returns an object of type [ScopeStartTriggerExportedOutputTerminal](#) representing the specified source [String](#).

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalGetHashCode Method

Gets the hash code for the current instance of ScopeStartTriggerExportedOutputTerminal.

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: Int32

Returns an Int32 representing the hash code for the current instance of ScopeStartTriggerExportedOutputTerminal.

See Also

Reference

ScopeStartTriggerExportedOutputTerminal Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalToString Method

Converts the current instance of ScopeStartTriggerExportedOutputTerminal to String.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeStartTriggerExportedOutputTerminal](#).

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)


[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Operators and Type Conversions

The [ScopeStartTriggerExportedOutputTerminal](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeStartTriggerExportedOutputTerminal object instances are equal.

	(String to ScopeStartTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeStartTriggerExportedOutputTerminal object.
	(ScopeStartTriggerExportedOutputTerminal to String)	Converts a ScopeStartTriggerExportedOutputTerminal object to an equivalent String .
	Inequality	Checks whether two ScopeStartTriggerExportedOutputTerminal object instances are not equal.

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquality

Checks whether two [ScopeStartTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeStartTriggerExportedOutputTerminal source1,
    ScopeStartTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeStartTriggerExportedOutputTerminal,
    source2 As ScopeStartTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
 Specifies the first [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
 Specifies the second [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalEquality Operator

Checks whether two [ScopeStartTriggerExportedOutputTerminal](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeStartTriggerExportedOutputTerminal source1,
    ScopeStartTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator = (
    source1 As ScopeStartTriggerExportedOutputTerminal,
    source2 As ScopeStartTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the first [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the second [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference





[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Conversion Operators

Overload List

	Name	Description
 	(String to ScopeStartTriggerExportedOutputTerminal)	Converts a specified String to an equivalent ScopeStartTriggerExportedOutputTerminal object.
 	(ScopeStartTriggerExportedOutputTerminal to String)	Converts a ScopeStartTriggerExportedOutputTerminal object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Conversion (String to ScopeStartTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeStartTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)


```
public static implicit operator ScopeStartTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeStartTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeStartTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

Returns the [ScopeStartTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Conversion (String to ScopeStartTriggerExportedOutputTerminal)

Converts a specified [String](#) to an equivalent [ScopeStartTriggerExportedOutputTerminal](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeStartTriggerExportedOutputTerminal (
    string source
)
```

```
Public Shared Widening Operator CType (
    source As String
) As ScopeStartTriggerExportedOutputTerminal
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeStartTriggerExportedOutputTerminal](#) object.

Return Value

Type: [ScopeStartTriggerExportedOutputTerminal](#)

Returns the [ScopeStartTriggerExportedOutputTerminal](#) object representing the specified [String](#).

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminal Conversion (ScopeStartTriggerExportedOutputTerminal to String)

Converts a [ScopeStartTriggerExportedOutputTerminal](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeStartTriggerExportedOutputTerminal source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeStartTriggerExportedOutputTerminal  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the [ScopeStartTriggerExportedOutputTerminal](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeStartTriggerExportedOutputTerminal](#) object.

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

Overload

[NationalInstruments.ModularInstruments.NIScope](#) Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeStartTriggerExportedOutputTerminalInequality Operator

Checks whether two [ScopeStartTriggerExportedOutputTerminal](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(
    ScopeStartTriggerExportedOutputTerminal source1,
    ScopeStartTriggerExportedOutputTerminal source2
)
```

```
Public Shared Operator <> (
    source1 As ScopeStartTriggerExportedOutputTerminal,
    source2 As ScopeStartTriggerExportedOutputTerminal
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the first [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeStartTriggerExportedOutputTerminal](#)
Specifies the second [ScopeStartTriggerExportedOutputTerminal](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeStartTriggerExportedOutputTerminal Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSubObject Class

Represents the base class for all configuration sub-objects. This is an internal class and is not intended for external use.

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

[More...](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax







[C#](#)/[VB](#)

```
public abstract class ScopeSubObject
```

```
Public MustInherit Class ScopeSubObject
```

The ScopeSubObject type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 MemberwiseClone	Creates a shallow copy of the current Object . (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

Contains members that are common to all sub-object NI-Scope classes. This is an internal class and is not intended for external use.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeAcquisition](#)

[NationalInstruments.ModularInstruments.NIScopeScopeAdvanceTrigger](#)







[NationalInstruments.ModularInstruments.NIScopeScopeArmReferenceTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeCalibration](#)
[NationalInstruments.ModularInstruments.NIScopeScopeCalibrationUtility](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannel](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelCollection](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelDdc](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelEqualizationFilter](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurement](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementFilter](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementReferenceLevel](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementTimeHistogram](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelMeasurementVoltageHistogram](#)
[NationalInstruments.ModularInstruments.NIScopeScopeChannelOnboardSignalProcessing](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeAdvanceTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeArmReferenceTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDigitalEdgeStartTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDriverIdentity](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDriverOperation](#)
[NationalInstruments.ModularInstruments.NIScopeScopeDriverUtility](#)
[NationalInstruments.ModularInstruments.NIScopeScopeEdgeTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)
[NationalInstruments.ModularInstruments.NIScopeScopeExternalCalibration](#)
[NationalInstruments.ModularInstruments.NIScopeScopeFractionalResample](#)
[NationalInstruments.ModularInstruments.NIScopeScopeGlitchTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeMeasurement](#)
[NationalInstruments.ModularInstruments.NIScopeScopeReferenceLevel](#)
[NationalInstruments.ModularInstruments.NIScopeScopeReferenceTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeRuntTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeSelfCalibration](#)
[NationalInstruments.ModularInstruments.NIScopeScopeStartTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeTiming](#)
[NationalInstruments.ModularInstruments.NIScopeScopeTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeTVTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeWidthTrigger](#)
[NationalInstruments.ModularInstruments.NIScopeScopeWindowReferenceTrigger](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSubObject Methods

The [ScopeSubObject](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 MemberwiseClone	Creates a shallow copy of the current Object . (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeSubObject Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeSubObject Methods

The [ScopeSubObject](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 MemberwiseClone	Creates a shallow copy of the current Object . (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeSubObject Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTiming Class

Represents the [ScopeMeasurement](#) sub-object that contains properties related to timing.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeTiming](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax









C#VB












```
public sealed class ScopeTiming : ScopeSubObject
```

```
Public NotInheritable Class ScopeTiming
    Inherits ScopeSubObject
```

The ScopeTiming type exposes the following members.

Properties

Name	Description
 AbsoluteSampleClockOffset	Gets or sets the absolute time offset of the sample clock relative to the reference clock.
 ClockSynchronizationPulseSource	Gets or sets the clock synchronization pulse source.
 EnforceRealtime	Gets or sets whether the digitizer enforces real-time measurements or allows equivalent-time measurements.
 FractionalResample	Gets the ScopeFractionalResample sub-object that you use to configure fractional resampling on the high speed digitizer.
 MoreRecordsThanMemoryAllowed	Gets or sets whether more records can be configured with NumberOfRecordsToAcquire than fit in the onboard memory.
 NumberOfRecordsToAcquire	Gets or sets the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this property to 1 indicates a single record acquisition.
 OnboardMemorySize	Gets the total combined amount of onboard memory for all channels in bytes.
 OscillatorPhaseDac	Gets or sets the binary phase DAC value that controls the delay added to the Phase Locked Loop (PLL) of the sample clock.

 PllLockStatus	Gets whether the PLL has remained locked/unlocked to the external reference clock since it was last checked.
 PollInterval	Gets or sets the poll interval to use during RIS acquisitions to check whether the acquisition is complete.
 RealtimeSampleRateMax	Gets the maximum real-time sample rate, in Hz.
 ReferenceClockRate	Gets or sets the frequency, in hertz, of the input, or reference clock, to which the internal sample clock timbase is synchronized if ReferenceClockSource is an external source.
 ReferenceClockSource	Gets or sets the input source for the PLL reference clock to which the digitizer will be phase-locked.
 RisAverages	Gets or sets the number of averages for each bin in an RIS acquisition.
 RisMethod	Gets or sets the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of RealtimeSampleRateMax .
 RisRateMax	Gets the maximum sample rate in RIS mode, in Hz.
 SampleClockTimebaseDivisor	Gets or sets the ratio between the SampleClockTimebaseRate and the SampleRate (which can be slower) if SampleClockTimebaseSource is an external source.
 SampleClockTimebaseMultiplier	If SampleClockTimebaseSource is an external source, this attribute specifies the ratio between the SampleClockTimebaseRate and the SampleRate , which can be higher. This property can be used in conjunction with the SampleClockTimebaseDivisor attribute. Some instruments use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by instrument and current configuration.
 SampleClockTimebaseRate	Gets or sets the frequency, in hertz, of the external clock used as the timebase source if SampleClockTimebaseSource is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.

	SampleClockTimebaseSource	Gets or sets the source of the sample clock timebase, which is the timebase used to control waveform sampling. The SampleRate may be the timebase itself or a divided version of the timebase, depending on the SampleRateMin (for internal sources) or the SampleClockTimebaseDivisor and SampleClockTimebaseMultiplier (for external sources).
--	---	--

[Top](#)

Methods

	Name	Description
	AdjustSampleClockRelativeDelay	Configures the relative sample clock delay in seconds when using the internal clock. Each time this method is called, the sample clock is delayed by the specified amount of time.
	ConfigureClock	Configures the properties for synchronizing the digitizer to a reference or sending the digitizer's reference clock output to be used as a synchronizing clock for other digitizers.
	ConfigureTiming	Configures the common properties of the horizontal subsystem for a multirecord acquisition in terms of minimum sample rate.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Horizontal attributes topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources







[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)














Copyright © National Instruments Corporation. All Rights Reserved.


ScopeTiming Properties

The [ScopeTiming](#) type exposes the following members.

Properties

Name	Description
 AbsoluteSampleClockOffset	Gets or sets the absolute time offset of the sample clock relative to the reference clock.
 ClockSynchronizationPulseSource	Gets or sets the clock synchronization pulse source.
 EnforceRealtime	Gets or sets whether the digitizer enforces real-time measurements or allows equivalent-time measurements.
 FractionalResample	Gets the ScopeFractionalResample sub-object that you use to configure fractional resampling on the high speed digitizer.
 MoreRecordsThanMemoryAllowed	Gets or sets whether more records can be configured with NumberOfRecordsToAcquire than fit in the onboard memory.
 NumberOfRecordsToAcquire	Gets or sets the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this property to 1 indicates a single record acquisition.

 OnboardMemorySize	Gets the total combined amount of onboard memory for all channels in bytes.
 OscillatorPhaseDac	Gets or sets the binary phase DAC value that controls the delay added to the Phase Locked Loop (PLL) of the sample clock.
 PllLockStatus	Gets whether the PLL has remained locked/unlocked to the external reference clock since it was last checked.
 PollInterval	Gets or sets the poll interval to use during RIS acquisitions to check whether the acquisition is complete.
 RealtimeSampleRateMax	Gets the maximum real-time sample rate, in Hz.
 ReferenceClockRate	Gets or sets the frequency, in hertz, of the input, or reference clock, to which the internal sample clock timbase is synchronized if ReferenceClockSource is an external source.
 ReferenceClockSource	Gets or sets the input source for the PLL reference clock to which the digitizer will be phase-locked.
 RisAverages	Gets or sets the number of averages for each bin in an RIS acquisition.
 RisMethod	Gets or sets the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of RealtimeSampleRateMax .
 RisRateMax	Gets the maximum sample rate in RIS mode, in Hz.
 SampleClockTimebaseDivisor	Gets or sets the ratio between the SampleClockTimebaseRate and the SampleRate (which can be slower) if SampleClockTimebaseSource is an external source.
 SampleClockTimebaseMultiplier	If SampleClockTimebaseSource is an external source, this attribute specifies the ratio between the SampleClockTimebaseRate and the SampleRate , which can be higher. This property can be used in conjunction with the SampleClockTimebaseDivisor attribute. Some instruments use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by instrument and current configuration.
 SampleClockTimebaseRate	Gets or sets the frequency, in hertz, of the external clock used as the timebase source if

		SampleClockTimebaseSource is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.
	SampleClockTimebaseSource	Gets or sets the source of the sample clock timebase, which is the timebase used to control waveform sampling. The SampleRate may be the timebase itself or a divided version of the timebase, depending on the SampleRateMin (for internal sources) or the SampleClockTimebaseDivisor and SampleClockTimebaseMultiplier (for external sources).

[Top](#)

See Also

Reference

[ScopeTiming Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAbsoluteSampleClockOffset Property

Gets or sets the absolute time offset of the sample clock relative to the reference clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan AbsoluteSampleClockOffset { get; set; }
```

```
Public Property AbsoluteSampleClockOffset As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The absolute time offset of the sample clock relative to the reference clock.

Exceptions

Exception	Condition
ObjectDisposedException	AbsoluteSampleClockOffset was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Configures the sample clock relationship with respect to the reference clock. This parameter is factored into NI-TClk adjustments and is typically used to improve the repeatability of NI-TClk Synchronization. When this parameter is read, the currently programmed value is returned. The range of the absolute sample clock offset is [-0.5 sample clock periods, 0.5 sample clock periods]. The default absolute sample clock offset is 0 s.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAbsoluteSampleClockOffset Property

Gets or sets the absolute time offset of the sample clock relative to the reference clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan AbsoluteSampleClockOffset { get; set; }
```

```
Public Property AbsoluteSampleClockOffset As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The absolute time offset of the sample clock relative to the reference clock.

Exceptions

Exception	Condition
ObjectDisposedException	AbsoluteSampleClockOffset was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Configures the sample clock relationship with respect to the reference clock. This parameter is factored into NI-TClk adjustments and is typically used to improve the repeatability of NI-TClk Synchronization. When this parameter is read, the currently programmed value is returned. The range of the absolute sample clock offset is [-0.5

sample clock periods, 0.5 sample clock periods]. The default absolute sample clock offset is 0 s.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingClockSynchronizationPulseSource Property

Gets or sets the clock synchronization pulse source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeClockSynchronizationPulseSource ClockSynchronizationPulseSource
{ get; set; }
```

```
Public Property ClockSynchronizationPulseSource As
ScopeClockSynchronizationPulseSource
    Get
    Set
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

The clock synchronization pulse source.

Exceptions

Exception	Condition
ObjectDisposedException	ClockSynchronizationPulseSource was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

This line should be the same for all instruments to be synchronized.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingEnforceRealtime Property

Gets or sets whether the digitizer enforces real-time measurements or allows equivalent-time measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool EnforceRealtime { get; set; }
```

```
Public Property EnforceRealtime As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is true.

Exceptions

Exception	Condition
InvalidOperationException	Trying to access an unset property.
ObjectDisposedException	The object has been disposed.
ModularInstrumentsException	The underlying driver returned an error.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingFractionalResample Property

Gets the [ScopeFractionalResample](#) sub-object that you use to configure fractional resampling on the high speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFractionalResample FractionalResample { get; }
```

```
Public ReadOnly Property FractionalResample As ScopeFractionalResample
    Get
```

Property Value

Type: [ScopeFractionalResample](#)

An object of type [ScopeFractionalResample](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingMoreRecordsThanMemoryAllowed Property

Gets or sets whether more records can be configured with [NumberOfRecordsToAcquire](#) than fit in the onboard memory.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool MoreRecordsThanMemoryAllowed { get; set; }
```

```
Public Property MoreRecordsThanMemoryAllowed As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

The default value is instrument-specific.

Remarks

If this property is set to , it is necessary to fetch records while the acquisition is in progress. Eventually, some of the records will be overwritten. The fetch method throws an exception if you attempt to fetch a record after it has been overwritten.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingNumberOfRecordsToAcquire Property

Gets or sets the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this property to 1 indicates a single record acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfRecordsToAcquire { get; set; }
```

```
Public Property NumberOfRecordsToAcquire As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

Number of records to acquire.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingOnboardMemorySize Property

Gets the total combined amount of onboard memory for all channels in bytes.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long OnboardMemorySize { get; }
```

```
Public ReadOnly Property OnboardMemorySize As Long  
    Get
```

Property Value

Type: [Int64](#)

Total combined amount of onboard memory for all channels in bytes.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingOscillatorPhaseDac Property

Gets or sets the binary phase DAC value that controls the delay added to the Phase Locked Loop (PLL) of the sample clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int OscillatorPhaseDac { get; set; }
```

```
Public Property OscillatorPhaseDac As Integer  
    Get  
    Set
```

Property Value

Type: [Int32](#)

Binary phase DAC value that controls the delay added to the PLL of the sample clock.

Remarks

If you set this value, sample clock adjust and TClk cannot do any sub-sample adjustment of the timebase sample clock.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingPllLockStatus Property

Gets whether the PLL has remained locked/unlocked to the external reference clock since it was last checked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool PllLockStatus { get; }
```

```
Public ReadOnly Property PllLockStatus As Boolean  
    Get
```

Property Value

Type: [Boolean](#)

true if the PLL has remained locked to the external reference clock since it was last checked and false, the PLL has become unlocked from the external reference clock since it was last checked.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingPollInterval Property

Gets or sets the poll interval to use during RIS acquisitions to check whether the acquisition is complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan PollInterval { get; set; }
```

```
Public Property PollInterval As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Poll interval, in PrecisionTimeSpan.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRealtimeSampleRateMax Property

Gets the maximum real-time sample rate, in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RealtimeSampleRateMax { get; }
```

```
Public ReadOnly Property RealtimeSampleRateMax As Double  
    Get
```

Property Value

Type: [Double](#)

Maximum real-time sample rate, in Hz.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingReferenceClockRate Property

Gets or sets the frequency, in hertz, of the input, or reference clock, to which the internal sample clock timbase is synchronized if [ReferenceClockSource](#) is an external source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ReferenceClockRate { get; set; }
```

```
Public Property ReferenceClockRate As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)
Frequency, in hertz

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingReferenceClockSource Property

Gets or sets the input source for the PLL reference clock to which the digitizer will be phase-locked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeInputClockSource ReferenceClockSource { get; set; }
```

```
Public Property ReferenceClockSource As ScopeInputClockSource  
    Get  
    Set
```

Property Value

Type: [ScopeInputClockSource](#)

An object of type [ScopeInputClockSource](#).

Remarks

The input source for the PLL reference clock is 1 MHz to 20 MHz clock on SMC-based digitizers

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisAverages Property

Gets or sets the number of averages for each bin in an RIS acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long RisAverages { get; set; }
```

```
Public Property RisAverages As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

The default value is instrument-specific.

Remarks

The number of averages times the oversampling factor is the minimum number of real-time acquisitions necessary to reconstruct the RIS waveform. Averaging is useful in RIS because the trigger times are not evenly spaced, so adjacent points in the reconstructed waveform cannot be accurately spaced. With averaging, the errors in both time and voltage are smoothed.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisMethod Property

Gets or sets the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of [RealtimeSampleRateMax](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeRisMethod RisMethod { get; set; }
```

```
Public Property RisMethod As ScopeRisMethod  
    Get  
    Set
```

Property Value

Type: [ScopeRisMethod](#)

An object of type [ScopeRisMethod](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisRateMax Property

Gets the maximum sample rate in RIS mode, in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RisRateMax { get; }
```

```
Public ReadOnly Property RisRateMax As Double  
    Get
```

Property Value

Type: [Double](#)

Maximum sample rate in RIS mode, in Hz.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseDivisor Property

Gets or sets the ratio between the [SampleClockTimebaseRate](#) and the [SampleRate](#) (which can be slower) if [SampleClockTimebaseSource](#) is an external source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long SampleClockTimebaseDivisor { get; set; }
```

```
Public Property SampleClockTimebaseDivisor As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

Ratio between sample clock timebase rate and the sample rate.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseMultiplier Property

If [SampleClockTimebaseSource](#) is an external source, this attribute specifies the ratio between the [SampleClockTimebaseRate](#) and the [SampleRate](#), which can be higher. This property can be used in conjunction with the [SampleClockTimebaseDivisor](#) attribute. Some instruments use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by instrument and current configuration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long SampleClockTimebaseMultiplier { get; set; }
```

```
Public Property SampleClockTimebaseMultiplier As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

If [SampleClockTimebaseSource](#) is an external source, this attribute specifies the ratio between the [SampleClockTimebaseRate](#) and the actual sample rate, which can be higher.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseRate Property

Gets or sets the frequency, in hertz, of the external clock used as the timebase source if [SampleClockTimebaseSource](#) is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleClockTimebaseRate { get; set; }
```

```
Public Property SampleClockTimebaseRate As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

Frequency, in hertz, of the external or internal clock.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseSource Property

Gets or sets the source of the sample clock timebase, which is the timebase used to control waveform sampling. The [SampleRate](#) may be the timebase itself or a divided version of the timebase, depending on the [SampleRateMin](#) (for internal sources) or the [SampleClockTimebaseDivisor](#) and [SampleClockTimebaseMultiplier](#) (for external sources).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleClockTimebaseSource SampleClockTimebaseSource { get; set; }
```

```
Public Property SampleClockTimebaseSource As ScopeSampleClockTimebaseSource  
    Get  
    Set
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

An object of the type [ScopeSampleClockTimebaseSource](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources







[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)














Copyright © National Instruments Corporation. All Rights Reserved.


ScopeTiming Properties

The [ScopeTiming](#) type exposes the following members.

Properties

Name	Description
 AbsoluteSampleClockOffset	Gets or sets the absolute time offset of the sample clock relative to the reference clock.
 ClockSynchronizationPulseSource	Gets or sets the clock synchronization pulse source.
 EnforceRealtime	Gets or sets whether the digitizer enforces real-time measurements or allows equivalent-time measurements.
 FractionalResample	Gets the ScopeFractionalResample sub-object that you use to configure fractional resampling on the high speed digitizer.
 MoreRecordsThanMemoryAllowed	Gets or sets whether more records can be configured with NumberOfRecordsToAcquire than fit in the onboard memory.
 NumberOfRecordsToAcquire	Gets or sets the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this property to 1 indicates a single record acquisition.

 OnboardMemorySize	Gets the total combined amount of onboard memory for all channels in bytes.
 OscillatorPhaseDac	Gets or sets the binary phase DAC value that controls the delay added to the Phase Locked Loop (PLL) of the sample clock.
 PllLockStatus	Gets whether the PLL has remained locked/unlocked to the external reference clock since it was last checked.
 PollInterval	Gets or sets the poll interval to use during RIS acquisitions to check whether the acquisition is complete.
 RealtimeSampleRateMax	Gets the maximum real-time sample rate, in Hz.
 ReferenceClockRate	Gets or sets the frequency, in hertz, of the input, or reference clock, to which the internal sample clock timbase is synchronized if ReferenceClockSource is an external source.
 ReferenceClockSource	Gets or sets the input source for the PLL reference clock to which the digitizer will be phase-locked.
 RisAverages	Gets or sets the number of averages for each bin in an RIS acquisition.
 RisMethod	Gets or sets the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of RealtimeSampleRateMax .
 RisRateMax	Gets the maximum sample rate in RIS mode, in Hz.
 SampleClockTimebaseDivisor	Gets or sets the ratio between the SampleClockTimebaseRate and the SampleRate (which can be slower) if SampleClockTimebaseSource is an external source.
 SampleClockTimebaseMultiplier	If SampleClockTimebaseSource is an external source, this attribute specifies the ratio between the SampleClockTimebaseRate and the SampleRate , which can be higher. This property can be used in conjunction with the SampleClockTimebaseDivisor attribute. Some instruments use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by instrument and current configuration.
 SampleClockTimebaseRate	Gets or sets the frequency, in hertz, of the external clock used as the timebase source if

		SampleClockTimebaseSource is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.
	SampleClockTimebaseSource	Gets or sets the source of the sample clock timebase, which is the timebase used to control waveform sampling. The SampleRate may be the timebase itself or a divided version of the timebase, depending on the SampleRateMin (for internal sources) or the SampleClockTimebaseDivisor and SampleClockTimebaseMultiplier (for external sources).

[Top](#)

See Also

Reference

[ScopeTiming Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAbsoluteSampleClockOffset Property

Gets or sets the absolute time offset of the sample clock relative to the reference clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public PrecisionTimeSpan AbsoluteSampleClockOffset { get; set; }
```

```
Public Property AbsoluteSampleClockOffset As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The absolute time offset of the sample clock relative to the reference clock.

Exceptions

Exception	Condition
ObjectDisposedException	AbsoluteSampleClockOffset was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Configures the sample clock relationship with respect to the reference clock. This parameter is factored into NI-TClk adjustments and is typically used to improve the repeatability of NI-TClk Synchronization. When this parameter is read, the currently programmed value is returned. The range of the absolute sample clock offset is [-0.5 sample clock periods, 0.5 sample clock periods]. The default absolute sample clock offset is 0 s.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAbsoluteSampleClockOffset Property

Gets or sets the absolute time offset of the sample clock relative to the reference clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan AbsoluteSampleClockOffset { get; set; }
```

```
Public Property AbsoluteSampleClockOffset As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The absolute time offset of the sample clock relative to the reference clock.

Exceptions

Exception	Condition
ObjectDisposedException	AbsoluteSampleClockOffset was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

Configures the sample clock relationship with respect to the reference clock. This parameter is factored into NI-TClk adjustments and is typically used to improve the repeatability of NI-TClk Synchronization. When this parameter is read, the currently programmed value is returned. The range of the absolute sample clock offset is [-0.5 sample clock periods, 0.5 sample clock periods]. The default absolute sample clock offset is 0 s.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingClockSynchronizationPulseSource Property

Gets or sets the clock synchronization pulse source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeClockSynchronizationPulseSource ClockSynchronizationPulseSource
{ get; set; }
```

```
Public Property ClockSynchronizationPulseSource As
ScopeClockSynchronizationPulseSource
    Get
    Set
```

Property Value

Type: [ScopeClockSynchronizationPulseSource](#)

The clock synchronization pulse source.

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	ClockSynchronizationPulseSource was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The underlying NI-SCOPE driver returned an error.

Remarks

This line should be the same for all instruments to be synchronized.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingEnforceRealtime Property

Gets or sets whether the digitizer enforces real-time measurements or allows equivalent-time measurements.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public bool EnforceRealtime { get; set; }
```

```
Public Property EnforceRealtime As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is true.

Exceptions

Exception	Condition
InvalidOperationException	Trying to access an unset property.
ObjectDisposedException	The object has been disposed.
ModularInstrumentsException	The underlying driver returned an error.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingFractionalResample Property

Gets the [ScopeFractionalResample](#) sub-object that you use to configure fractional resampling on the high speed digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeFractionalResample FractionalResample { get; }
```

```
Public ReadOnly Property FractionalResample As ScopeFractionalResample
    Get
```

Property Value

Type: [ScopeFractionalResample](#)

An object of type [ScopeFractionalResample](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingMoreRecordsThanMemoryAllowed Property

Gets or sets whether more records can be configured with [NumberOfRecordsToAcquire](#) than fit in the onboard memory.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool MoreRecordsThanMemoryAllowed { get; set; }
```

```
Public Property MoreRecordsThanMemoryAllowed As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

The default value is instrument-specific.

Remarks

If this property is set to `True`, it is necessary to fetch records while the acquisition is in progress. Eventually, some of the records will be overwritten. The fetch method throws an exception if you attempt to fetch a record after it has been overwritten.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingNumberOfRecordsToAcquire Property

Gets or sets the number of records to acquire. Can be used for multirecord acquisitions and single record acquisitions. Setting this property to 1 indicates a single record acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long NumberOfRecordsToAcquire { get; set; }
```

```
Public Property NumberOfRecordsToAcquire As Long  
    Get  
    Set
```

Property Value

Type: [Int64](#)

Number of records to acquire.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingOnboardMemorySize Property

Gets the total combined amount of onboard memory for all channels in bytes.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long OnboardMemorySize { get; }
```

```
Public ReadOnly Property OnboardMemorySize As Long  
    Get
```

Property Value

Type: [Int64](#)

Total combined amount of onboard memory for all channels in bytes.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingOscillatorPhaseDac Property

Gets or sets the binary phase DAC value that controls the delay added to the Phase Locked Loop (PLL) of the sample clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public int OscillatorPhaseDac { get; set; }
```

```
Public Property OscillatorPhaseDac As Integer  
    Get  
    Set
```

Property Value

Type: [Int32](#)

Binary phase DAC value that controls the delay added to the PLL of the sample clock.

Remarks

If you set this value, sample clock adjust and TClk cannot do any sub-sample adjustment of the timebase sample clock.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingPllLockStatus Property

Gets whether the PLL has remained locked/unlocked to the external reference clock since it was last checked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool PllLockStatus { get; }
```

```
Public ReadOnly Property PllLockStatus As Boolean  
    Get
```

Property Value

Type: [Boolean](#)

true if the PLL has remained locked to the external reference clock since it was last checked and false, the PLL has become unlocked from the external reference clock since it was last checked.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingPollInterval Property

Gets or sets the poll interval to use during RIS acquisitions to check whether the acquisition is complete.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan PollInterval { get; set; }
```

```
Public Property PollInterval As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Poll interval, in PrecisionTimeSpan.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRealtimeSampleRateMax Property

Gets the maximum real-time sample rate, in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RealtimeSampleRateMax { get; }
```

```
Public ReadOnly Property RealtimeSampleRateMax As Double  
    Get
```

Property Value

Type: [Double](#)

Maximum real-time sample rate, in Hz.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingReferenceClockRate Property

Gets or sets the frequency, in hertz, of the input, or reference clock, to which the internal sample clock timbase is synchronized if [ReferenceClockSource](#) is an external source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ReferenceClockRate { get; set; }
```

```
Public Property ReferenceClockRate As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

Frequency, in hertz

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingReferenceClockSource Property

Gets or sets the input source for the PLL reference clock to which the digitizer will be phase-locked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeInputClockSource ReferenceClockSource { get; set; }
```

```
Public Property ReferenceClockSource As ScopeInputClockSource
    Get
    Set
```

Property Value

Type: [ScopeInputClockSource](#)

An object of type [ScopeInputClockSource](#).

Remarks

The input source for the PLL reference clock is 1 MHz to 20 MHz clock on SMC-based digitizers

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisAverages Property

Gets or sets the number of averages for each bin in an RIS acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public long RisAverages { get; set; }
```

```
Public Property RisAverages As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The default value is instrument-specific.

Remarks

The number of averages times the oversampling factor is the minimum number of real-time acquisitions necessary to reconstruct the RIS waveform. Averaging is useful in RIS because the trigger times are not evenly spaced, so adjacent points in the reconstructed waveform cannot be accurately spaced. With averaging, the errors in both time and voltage are smoothed.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisMethod Property

Gets or sets the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the value of [RealtimeSampleRateMax](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeRisMethod RisMethod { get; set; }
```

```
Public Property RisMethod As ScopeRisMethod  
    Get  
    Set
```

Property Value

Type: [ScopeRisMethod](#)

An object of type [ScopeRisMethod](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingRisRateMax Property

Gets the maximum sample rate in RIS mode, in Hz.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RisRateMax { get; }
```

```
Public ReadOnly Property RisRateMax As Double
    Get
```

Property Value

Type: [Double](#)

Maximum sample rate in RIS mode, in Hz.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseDivisor Property

Gets or sets the ratio between the [SampleClockTimebaseRate](#) and the [SampleRate](#) (which can be slower) if [SampleClockTimebaseSource](#) is an external source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long SampleClockTimebaseDivisor { get; set; }
```

```
Public Property SampleClockTimebaseDivisor As Long
    Get
    Set
```


Property Value

Type: [Int64](#)

Ratio between sample clock timebase rate and the sample rate.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseMultiplier Property

If [SampleClockTimebaseSource](#) is an external source, this attribute specifies the ratio between the [SampleClockTimebaseRate](#) and the [SampleRate](#), which can be higher. This property can be used in conjunction with the [SampleClockTimebaseDivisor](#) attribute. Some instruments use multiple ADCs to sample the same channel at an effective sample rate that is greater than the specified clock rate. When providing an external sample clock use this property to indicate when you want a higher sample rate. Valid values for this attribute vary by instrument and current configuration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long SampleClockTimebaseMultiplier { get; set; }
```

```
Public Property SampleClockTimebaseMultiplier As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

If [SampleClockTimebaseSource](#) is an external source, this attribute specifies the ratio between the [SampleClockTimebaseRate](#) and the actual sample rate, which can be higher.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseRate Property

Gets or sets the frequency, in hertz, of the external clock used as the timebase source if [SampleClockTimebaseSource](#) is an external source or, if the timebase source is the internal clock for oscilloscopes that support multiple onboard clock rates, of the internal clock.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double SampleClockTimebaseRate { get; set; }
```

```
Public Property SampleClockTimebaseRate As Double
    Get
    Set
```

Property Value

Type: [Double](#)

Frequency, in hertz, of the external or internal clock.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingSampleClockTimebaseSource Property

Gets or sets the source of the sample clock timebase, which is the timebase used to control waveform sampling. The [SampleRate](#) may be the timebase itself or a divided version of the timebase, depending on the [SampleRateMin](#) (for internal sources) or the [SampleClockTimebaseDivisor](#) and [SampleClockTimebaseMultiplier](#) (for external sources).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeSampleClockTimebaseSource SampleClockTimebaseSource { get; set; }
```

```
Public Property SampleClockTimebaseSource As ScopeSampleClockTimebaseSource
    Get
    Set
```

Property Value

Type: [ScopeSampleClockTimebaseSource](#)

An object of the type [ScopeSampleClockTimebaseSource](#).

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources




[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTiming Methods

The [ScopeTiming](#) type exposes the following members.

Methods

Name	Description
 AdjustSampleClockRelativeDelay	Configures the relative sample clock delay in seconds when using the internal clock. Each time this method is called, the sample clock is delayed by the specified amount of time.
 ConfigureClock	Configures the properties for synchronizing the digitizer to a reference or sending the digitizer's reference clock output to be used as a synchronizing clock for other digitizers.
 ConfigureTiming	Configures the common properties of the horizontal subsystem for a multirecord acquisition in terms of minimum sample rate.

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeTiming Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAdjustSampleClockRelativeDelay Method

Configures the relative sample clock delay in seconds when using the internal clock. Each time this method is called, the sample clock is delayed by the specified amount of time.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public void AdjustSampleClockRelativeDelay(
    PrecisionTimeSpan delay
)
```

```
Public Sub AdjustSampleClockRelativeDelay (  
    delay As PrecisionTimeSpan  
)
```

Parameters

delay

Type: `NationalInstrumentsPrecisionTimeSpan`

The amount of time, in seconds, to delay the sample clock. This is a relative value, so repeated calls to this method delays the sample clock by this amount every time.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingAdjustSampleClockRelativeDelay Method

Configures the relative sample clock delay in seconds when using the internal clock. Each time this method is called, the sample clock is delayed by the specified amount of time.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

[C#VB](#)

```
public void AdjustSampleClockRelativeDelay(
    PrecisionTimeSpan delay
)
```

```
Public Sub AdjustSampleClockRelativeDelay (
    delay As PrecisionTimeSpan
)
```

Parameters

delay

Type: `NationalInstrumentsPrecisionTimeSpan`

The amount of time, in seconds, to delay the sample clock. This is a relative value, so repeated calls to this method delays the sample clock by this amount every time.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingConfigureClock Method

Configures the properties for synchronizing the digitizer to a reference or sending the digitizer's reference clock output to be used as a synchronizing clock for other digitizers.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public void ConfigureClock(
    ScopeInputClockSource inputClockSource,
    ScopeOutputClockSource outputClockSource,
    ScopeClockSynchronizationPulseSource clockSynchronizationPulseSource,
    bool masterEnabled
)
```

```
Public Sub ConfigureClock (
    inputClockSource As ScopeInputClockSource,
    outputClockSource As ScopeOutputClockSource,
    clockSynchronizationPulseSource As ScopeClockSynchronizationPulseSource,
    masterEnabled As Boolean
)
```

Parameters

inputClockSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeInputClockSource](#)
The input source for the reference clock to which the 100 MHz sample clock is phase-locked.

outputClockSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeOutputClockSource](#)
The output source for the reference clock to which another scope's sample clock can be phased-locked.

clockSynchronizationPulseSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeClockSynchronizationPulseSource](#)
This line should be the same for all instruments to be synchronized.

masterEnabled

Type: [SystemBoolean](#)
Specifies whether you want the instrument to be a master or a slave. The master instrument is typically the originator of the trigger signal and clock sync pulse. For a standalone instrument, set this property to VI_FALSE.

Remarks

Some features are not supported by all digitizers. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTimingConfigureTiming Method

Configures the common properties of the horizontal subsystem for a multirecord acquisition in terms of minimum sample rate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ConfigureTiming(  
    double sampleRateMin,  
    int numberOfPointsMin,  
    double referencePosition,  
    int numberOfRecords,  
    bool enforceRealtime  
)
```

```
Public Sub ConfigureTiming (  
    sampleRateMin As Double,  
    numberOfPointsMin As Integer,  
    referencePosition As Double,  
    numberOfRecords As Integer,
```

```

        enforceRealtime As Boolean
    )

```

Parameters

sampleRateMin

Type: [SystemDouble](#)

The sampling rate for the acquisition.

numberOfPointsMin

Type: [SystemInt32](#)

The minimum number of points you need in the record for each channel

referencePosition

Type: [SystemDouble](#)

The position of the reference event in the waveform record specified as a percentage.

numberOfRecords

Type: [SystemInt32](#)

The number of records to acquire.

enforceRealtime

Type: [SystemBoolean](#)

Indicates whether the digitizer enforces real-time measurements or allows equivalent-time (RIS) measurements. Not all digitizers support RIS. Refer to [Features Supported by Device](#) for more information.

See Also

Reference

[ScopeTiming Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger Class

Represents the [ScopeMeasurement](#) sub-object that contains the properties and methods related to triggering in NI-SCOPE.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject

NationalInstruments.ModularInstruments.NIScopeScopeTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax








C#VB














```
public sealed class ScopeTrigger : ScopeSubObject,
    IIVI_ScopeTrigger
```

```
Public NotInheritable Class ScopeTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTrigger
```

The ScopeTrigger type exposes the following members.






Properties





Name	Description
 AdvanceTrigger	Gets the ScopeAdvanceTrigger sub-object.
 ArmReferenceTrigger	Gets the ScopeArmReferenceTrigger sub-object.
 Coupling	Gets or sets how the digitizer couples the trigger source.
 EdgeTrigger	Gets the ScopeEdgeTrigger sub-object.
 EndOfRecordToAdvanceTriggerHoldoff	Gets or sets the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.
 ExternalTriggerSourceImpedance	Gets or sets the input impedance, in Ohms, for the external analog trigger channel.
 GlitchTrigger	Gets the ScopeGlitchTrigger sub-object.

 Holdoff	Gets or sets the length of time, in seconds, the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger.
 IsAutoTriggered	Gets whether the acquisition was auto triggered.
 Level	Gets or sets the voltage threshold, in volts, for the trigger subsystem.
 Modifier	Configures an instrument to automatically complete an acquisition if a trigger has not been received.
 ReferenceTrigger	Gets the ScopeReferenceTrigger sub-object.
 RuntTrigger	Gets the ScopeRuntTrigger sub-object.
 Source	Gets or sets the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.
 StartToReferenceTriggerHoldoff	Gets or sets the length of time, in seconds, you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a Reference Stop trigger.
 StartTrigger	Gets the ScopeStartTrigger sub-object.
 TdcEnabled	Gets or sets whether the digitizer should record the trigger position precisely using time-digital conversion (TDC).
 TV	Gets the ScopeTVTrigger sub-object.
 Type	Get or sets the type of the trigger in use.
 WidthTrigger	Gets the ScopeWidthTrigger sub-object.

[Top](#)

Methods

Name	Description
 ConfigureTriggerDigital	Configures the common properties of a digital trigger.
 ConfigureTriggerHysteresis	Configures common properties for analog hysteresis triggering.
 ConfigureTriggerImmediate	Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.
 ConfigureTriggerSoftware	Configures common properties for software triggering.
 ConfigureTriggerWindow	Configures common properties for analog window triggering.

 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the [Triggering functions and attributes topics in the NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources



















[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger Properties

The [ScopeTrigger](#) type exposes the following members.

Properties

Name	Description
 AdvanceTrigger	Gets the ScopeAdvanceTrigger sub-object.
 ArmReferenceTrigger	Gets the ScopeArmReferenceTrigger sub-object.
 Coupling	Gets or sets how the digitizer couples the trigger source.
 EdgeTrigger	Gets the ScopeEdgeTrigger sub-object.
 EndOfRecordToAdvanceTriggerHoldoff	Gets or sets the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.
 ExternalTriggerSourceImpedance	Gets or sets the input impedance, in Ohms, for the external analog trigger channel.
 GlitchTrigger	Gets the ScopeGlitchTrigger sub-object.
 Holdoff	Gets or sets the length of time, in seconds, the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger.
 IsAutoTriggered	Gets whether the acquisition was auto triggered.
 Level	Gets or sets the voltage threshold, in volts, for the trigger subsystem.
 Modifier	Configures an instrument to automatically complete an acquisition if a trigger has not been received.
 ReferenceTrigger	Gets the ScopeReferenceTrigger sub-object.
 RuntTrigger	Gets the ScopeRuntTrigger sub-object.
 Source	Gets or sets the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.
 StartToReferenceTriggerHoldoff	Gets or sets the length of time, in seconds, you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a Reference Stop trigger.
 StartTrigger	Gets the ScopeStartTrigger sub-object.
 TdcEnabled	Gets or sets whether the digitizer should record the trigger position precisely using time-digital conversion (TDC).
 TV	Gets the ScopeTVTrigger sub-object.

 Type	Get or sets the type of the trigger in use.
 WidthTrigger	Gets the ScopeWidthTrigger sub-object.

[Top](#)

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerAdvanceTrigger Property

Gets the [ScopeAdvanceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeAdvanceTrigger AdvanceTrigger { get; }
```

```
Public ReadOnly Property AdvanceTrigger As ScopeAdvanceTrigger
    Get
```

Property Value

Type: [ScopeAdvanceTrigger](#)

An object of the type [ScopeAdvanceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerAdvanceTrigger Property

Gets the [ScopeAdvanceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTrigger AdvanceTrigger { get; }
```

```
Public ReadOnly Property AdvanceTrigger As ScopeAdvanceTrigger  
    Get
```

Property Value

Type: [ScopeAdvanceTrigger](#)

An object of the type [ScopeAdvanceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerArmReferenceTrigger Property

Gets the [ScopeArmReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTrigger ArmReferenceTrigger { get; }
```

```
Public ReadOnly Property ArmReferenceTrigger As ScopeArmReferenceTrigger  
    Get
```

Property Value

Type: [ScopeArmReferenceTrigger](#)
An object of the type [ScopeArmReferenceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerCoupling Property

Gets or sets how the digitizer couples the trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerCoupling Coupling { get; set; }
```

```
Public Property Coupling As ScopeTriggerCoupling
    Get
    Set
```

Property Value

Type: [ScopeTriggerCoupling](#)
Specifies the [ScopeTriggerCoupling](#).

Remarks

This property affects instrument operation only when [ScopeTriggerType](#) is set to [Edge](#), [Hysteresis](#), or [Window](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerEdgeTrigger Property

Gets the [ScopeEdgeTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEdgeTrigger EdgeTrigger { get; }
```

```
Public ReadOnly Property EdgeTrigger As ScopeEdgeTrigger
    Get
```

Property Value

Type: [ScopeEdgeTrigger](#)

An object of the type [ScopeEdgeTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerEndOfRecordToAdvanceTriggerHoldoff

Gets or sets the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan EndOfRecordToAdvanceTriggerHoldoff { get; set; }
```

```
Public Property EndOfRecordToAdvanceTriggerHoldoff As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

The adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.

Exceptions

Exception	Condition
ObjectDisposedException	EndOfRecordToAdvanceTriggerHoldoff was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

Remarks

This attribute disables sampling and triggering.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerExternalTriggerSourceImpedance Property

Gets or sets the input impedance, in Ohms, for the external analog trigger channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ExternalTriggerSourceImpedance { get; set; }
```

```
Public Property ExternalTriggerSourceImpedance As Double
    Get
    Set
```

Property Value

Type: [Double](#)

Valid values are 50 Ohms and 1 MOhm.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerGlitchTrigger Property

Gets the [ScopeGlitchTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeGlitchTrigger GlitchTrigger { get; }
```

```
Public ReadOnly Property GlitchTrigger As ScopeGlitchTrigger
    Get
```

Property Value

Type: [ScopeGlitchTrigger](#)

An object of the type [ScopeGlitchTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	GlitchTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerHoldoff Property

Gets or sets the length of time, in seconds, the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public PrecisionTimeSpan Holdoff { get; set; }
```

```
Public Property Holdoff As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

Remarks

This property affects instrument operation only when the digitizer requires multiple acquisitions to build a complete waveform. The digitizer requires multiple waveform acquisitions when it uses equivalent-time sampling or when the digitizer is configured for a multirecord acquisition through the property [NumberOfRecordsToAcquire](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerIsAutoTriggered Property

Gets whether the acquisition was auto triggered.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool IsAutoTriggered { get; }
```

```
Public ReadOnly Property IsAutoTriggered As Boolean
    Get
```

Property Value

Type: [Boolean](#)

Auto triggering occurs if the [Modifier](#) is set to [Auto](#), and no trigger has been received for a certain amount of time. , if the acquisition is triggered in this scenario and , if the acquisition is not triggered.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerLevel Property

Gets or sets the voltage threshold, in volts, for the trigger subsystem.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Level { get; set; }
```

```
Public Property Level As Double
    Get
    Set
```


Property Value

Type: [Double](#)

The values of the properties [Range](#) and [Offset](#) determine the valid range for the trigger level on the channel used as the Trigger Source. The value passed for the parameter must meet the following conditions:

Trigger Level \leq (Range/2) + Offset

Range \geq (-Range/2) + Offset

Implements

[IIVI_ScopeTriggerLevel](#)

Remarks

This property affects instrument behavior only when the [Type](#) is set to [Edge](#), [Hysteresis](#), or [Window](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerModifier Property

Configures an instrument to automatically complete an acquisition if a trigger has not been received.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

[C#](#)[VB](#)

```
public ScopeTriggerModifier Modifier { get; set; }
```

```
Public Property Modifier As ScopeTriggerModifier  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerModifier](#)

Specifies the [ScopeTriggerModifier](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerReferenceTrigger Property

Gets the [ScopeReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTrigger ReferenceTrigger { get; }
```

```
Public ReadOnly Property ReferenceTrigger As ScopeReferenceTrigger
    Get
```

Property Value

Type: [ScopeReferenceTrigger](#)

An object of the type [ScopeReferenceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerRuntTrigger Property

Gets the [ScopeRuntTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeRuntTrigger RuntTrigger { get; }
```

```
Public ReadOnly Property RuntTrigger As ScopeRuntTrigger
    Get
```

Property Value

Type: [ScopeRuntTrigger](#)

An object of the type [ScopeRuntTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	RuntTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Property

Gets or sets the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeTriggerSource
    Get
    Set
```

Property Value

Type: [ScopeTriggerSource](#)

An object of the type [ScopeTriggerSource](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerStartToReferenceTriggerHoldoff Property

Gets or sets the length of time, in seconds, you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a Reference Stop trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public PrecisionTimeSpan StartToReferenceTriggerHoldoff { get; set; }
```

```
Public Property StartToReferenceTriggerHoldoff As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerStartTrigger Property

Gets the [ScopeStartTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeStartTrigger StartTrigger { get; }
```

```
Public ReadOnly Property StartTrigger As ScopeStartTrigger  
    Get
```

Property Value

Type: [ScopeStartTrigger](#)

An object of the type [ScopeStartTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerTdcEnabled Property

Gets or sets whether the digitizer should record the trigger position precisely using time-digital conversion (TDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool TdcEnabled { get; set; }
```

```
Public Property TdcEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

Gets or sets whether the digitizer should record the trigger position precisely using TDC.

Remarks

Disabling TDC by setting this property to may reduce rearm time but causes the digitizer to round the trigger position to the nearest sample clock.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerTV Property

Gets the [ScopeTVTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public ScopeTVTrigger TV { get; }
```

```
Public ReadOnly Property TV As ScopeTVTrigger  
    Get
```

Property Value

Type: [ScopeTVTrigger](#)

An object of the type [ScopeTVTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerType Property

Get or sets the type of the trigger in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerType Type { get; set; }
```

```
Public Property Type As ScopeTriggerType  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerType](#)

Specifies the [ScopeTriggerType](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerWidthTrigger Property

Gets the [ScopeWidthTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWidthTrigger WidthTrigger { get; }
```

```
Public ReadOnly Property WidthTrigger As ScopeWidthTrigger
    Get
```

Property Value

Type: [ScopeWidthTrigger](#)

An object of the type [ScopeWidthTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	WidthTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)


















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger Properties

The [ScopeTrigger](#) type exposes the following members.

Properties

Name	Description
 AdvanceTrigger	Gets the ScopeAdvanceTrigger sub-object.
 ArmReferenceTrigger	Gets the ScopeArmReferenceTrigger sub-object.
 Coupling	Gets or sets how the digitizer couples the trigger source.

 EdgeTrigger	Gets the ScopeEdgeTrigger sub-object.
 EndOfRecordToAdvanceTriggerHoldoff	Gets or sets the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.
 ExternalTriggerSourceImpedance	Gets or sets the input impedance, in Ohms, for the external analog trigger channel.
 GlitchTrigger	Gets the ScopeGlitchTrigger sub-object.
 Holdoff	Gets or sets the length of time, in seconds, the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger.
 IsAutoTriggered	Gets whether the acquisition was auto triggered.
 Level	Gets or sets the voltage threshold, in volts, for the trigger subsystem.
 Modifier	Configures an instrument to automatically complete an acquisition if a trigger has not been received.
 ReferenceTrigger	Gets the ScopeReferenceTrigger sub-object.
 RuntTrigger	Gets the ScopeRuntTrigger sub-object.
 Source	Gets or sets the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.
 StartToReferenceTriggerHoldoff	Gets or sets the length of time, in seconds, you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a Reference Stop trigger.
 StartTrigger	Gets the ScopeStartTrigger sub-object.
 TdcEnabled	Gets or sets whether the digitizer should record the trigger position precisely using time-digital conversion (TDC).
 TV	Gets the ScopeTVTrigger sub-object.
 Type	Get or sets the type of the trigger in use.
 WidthTrigger	Gets the ScopeWidthTrigger sub-object.

[Top](#)

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerAdvanceTrigger Property

Gets the [ScopeAdvanceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTrigger AdvanceTrigger { get; }
```

```
Public ReadOnly Property AdvanceTrigger As ScopeAdvanceTrigger  
    Get
```

Property Value

Type: [ScopeAdvanceTrigger](#)

An object of the type [ScopeAdvanceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerAdvanceTrigger Property

Gets the [ScopeAdvanceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeAdvanceTrigger AdvanceTrigger { get; }
```

```
Public ReadOnly Property AdvanceTrigger As ScopeAdvanceTrigger  
    Get
```

Property Value

Type: [ScopeAdvanceTrigger](#)

An object of the type [ScopeAdvanceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerArmReferenceTrigger Property

Gets the [ScopeArmReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeArmReferenceTrigger ArmReferenceTrigger { get; }
```

```
Public ReadOnly Property ArmReferenceTrigger As ScopeArmReferenceTrigger
    Get
```

Property Value

Type: [ScopeArmReferenceTrigger](#)

An object of the type [ScopeArmReferenceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerCoupling Property

Gets or sets how the digitizer couples the trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerCoupling Coupling { get; set; }
```

```
Public Property Coupling As ScopeTriggerCoupling
    Get
    Set
```

Property Value

Type: [ScopeTriggerCoupling](#)

Specifies the [ScopeTriggerCoupling](#).

Remarks

This property affects instrument operation only when [ScopeTriggerType](#) is set to [Edge](#), [Hysteresis](#), or [Window](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerEdgeTrigger Property

Gets the [ScopeEdgeTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeEdgeTrigger EdgeTrigger { get; }
```

```
Public ReadOnly Property EdgeTrigger As ScopeEdgeTrigger  
    Get
```

Property Value

Type: [ScopeEdgeTrigger](#)

An object of the type [ScopeEdgeTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerEndOfRecordToAdvanceTriggerHoldoff Property

Gets or sets the adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan EndOfRecordToAdvanceTriggerHoldoff { get; set; }
```

```
Public Property EndOfRecordToAdvanceTriggerHoldoff As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The adjustable period of time between records that allows for fetching data from the digitizer during a multi-record acquisition.

Exceptions

Exception	Condition
ObjectDisposedException	EndOfRecordToAdvanceTriggerHoldoff was accessed after associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The underlying NI-SCOPE driver returned error.

Remarks

This attribute disables sampling and triggering.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerExternalTriggerSourceImpedance Property

Gets or sets the input impedance, in Ohms, for the external analog trigger channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double ExternalTriggerSourceImpedance { get; set; }
```

```
Public Property ExternalTriggerSourceImpedance As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

Valid values are 50 Ohms and 1 MOhm.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerGlitchTrigger Property

Gets the [ScopeGlitchTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeGlitchTrigger GlitchTrigger { get; }
```

```
Public ReadOnly Property GlitchTrigger As ScopeGlitchTrigger  
    Get
```

Property Value

Type: [ScopeGlitchTrigger](#)

An object of the type [ScopeGlitchTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	GlitchTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerHoldoff Property

Gets or sets the length of time, in seconds, the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan Holdoff { get; set; }
```

```
Public Property Holdoff As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

Remarks

This property affects instrument operation only when the digitizer requires multiple acquisitions to build a complete waveform. The digitizer requires multiple waveform acquisitions when it uses equivalent-time sampling or when the digitizer

is configured for a multirecord acquisition through the property [NumberOfRecordsToAcquire](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger.IsAutoTriggered Property

Gets whether the acquisition was auto triggered.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool IsAutoTriggered { get; }
```

```
Public ReadOnly Property IsAutoTriggered As Boolean  
    Get
```

Property Value

Type: [Boolean](#)

Auto triggering occurs if the [Modifier](#) is set to [Auto](#), and no trigger has been received for a certain amount of time. , if the acquisition is triggered in this scenario and , if the acquisition is not triggered.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerLevel Property

Gets or sets the voltage threshold, in volts, for the trigger subsystem.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Level { get; set; }
```

```
Public Property Level As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The values of the properties [Range](#) and [Offset](#) determine the valid range for the trigger level on the channel used as the Trigger Source. The value passed for the parameter must meet the following conditions:

Trigger Level \leq (Range/2) + Offset

Range \geq (-Range/2) + Offset

Implements

IlviScopeTriggerLevel

Remarks

This property affects instrument behavior only when the [Type](#) is set to [Edge](#), [Hysteresis](#), or [Window](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerModifier Property

Configures an instrument to automatically complete an acquisition if a trigger has not been received.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerModifier Modifier { get; set; }
```

```
Public Property Modifier As ScopeTriggerModifier  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerModifier](#)

Specifies the [ScopeTriggerModifier](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerReferenceTrigger Property

Gets the [ScopeReferenceTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeReferenceTrigger ReferenceTrigger { get; }
```

```
Public ReadOnly Property ReferenceTrigger As ScopeReferenceTrigger  
    Get
```

Property Value

Type: [ScopeReferenceTrigger](#)

An object of the type [ScopeReferenceTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerRuntTrigger Property

Gets the [ScopeRuntTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeRuntTrigger RuntTrigger { get; }
```

```
Public ReadOnly Property RuntTrigger As ScopeRuntTrigger
    Get
```

Property Value

Type: [ScopeRuntTrigger](#)

An object of the type [ScopeRuntTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	RuntTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Property

Gets or sets the source the digitizer monitors for the trigger event. Trigger sources vary for each digitizer model.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerSource Source { get; set; }
```

```
Public Property Source As ScopeTriggerSource  
    Get  
    Set
```

Property Value

Type: [ScopeTriggerSource](#)

An object of the type [ScopeTriggerSource](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerStartToReferenceTriggerHoldoff Property

Gets or sets the length of time, in seconds, you want the digitizer to wait after it starts acquiring data until the digitizer enables the trigger system to detect a Reference Stop trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan StartToReferenceTriggerHoldoff { get; set; }
```

```
Public Property StartToReferenceTriggerHoldoff As PrecisionTimeSpan  
    Get  
    Set
```

Property Value

Type: PrecisionTimeSpan

Valid values for this property are instrument-dependent, and range from 0.0 seconds to 171.8 seconds.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerStartTrigger Property

Gets the [ScopeStartTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeStartTrigger StartTrigger { get; }
```

```
Public ReadOnly Property StartTrigger As ScopeStartTrigger
    Get
```

Property Value

Type: [ScopeStartTrigger](#)

An object of the type [ScopeStartTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerTdcEnabled Property

Gets or sets whether the digitizer should record the trigger position precisely using time-digital conversion (TDC).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool TdcEnabled { get; set; }
```

```
Public Property TdcEnabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

Gets or sets whether the digitizer should record the trigger position precisely using TDC.

Remarks

Disabling TDC by setting this property to may reduce rearm time but causes the digitizer to round the trigger position to the nearest sample clock.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerTV Property

Gets the [ScopeTVTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTrigger TV { get; }
```

```
Public ReadOnly Property TV As ScopeTVTrigger
    Get
```

Property Value

Type: [ScopeTVTrigger](#)

An object of the type [ScopeTVTrigger](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerType Property

Get or sets the type of the trigger in use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTriggerType Type { get; set; }
```

```
Public Property Type As ScopeTriggerType
    Get
    Set
```

Property Value

Type: [ScopeTriggerType](#)

Specifies the [ScopeTriggerType](#).

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger.WidthTrigger Property

Gets the [ScopeWidthTrigger](#) sub-object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWidthTrigger WidthTrigger { get; }
```

```
Public ReadOnly Property WidthTrigger As ScopeWidthTrigger  
    Get
```

Property Value

Type: [ScopeWidthTrigger](#)

An object of the type [ScopeWidthTrigger](#).

Exceptions

Exception	Condition
ObjectDisposedException	WidthTrigger was called after the associated NIScope or ScopeDriverUtility object was disposed.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTrigger Methods

The [ScopeTrigger](#) type exposes the following members.

Methods

Name	Description
ConfigureTriggerDigital	Configures the common properties of a digital trigger.
ConfigureTriggerHysteresis	Configures common properties for analog hysteresis triggering.
ConfigureTriggerImmediate	Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.
ConfigureTriggerSoftware	Configures common properties for software triggering.
ConfigureTriggerWindow	Configures common properties for analog window triggering.
Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
GetHashCode	Serves as the default hash function. (Inherited from Object .)
GetType	Gets the Type of the current instance. (Inherited from Object .)
ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerDigital Method

Configures the common properties of a digital trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void ConfigureTriggerDigital (
    ScopeTriggerSource triggerSource,
    ScopeTriggerSlope triggerSlope,
    PrecisionTimeSpan triggerHoldoff,
    PrecisionTimeSpan triggerDelay
)
```

```
Public Sub ConfigureTriggerDigital (
    triggerSource As ScopeTriggerSource,
    triggerSlope As ScopeTriggerSlope,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerSlope

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSlope](#)

A value indicating whether you want a rising edge or a falling edge to trigger the digitizer.

triggerHoldoff

Type: `NationalInstrumentsPrecisionTimeSpan`

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: `NationalInstrumentsPrecisionTimeSpan`

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

See Also

Reference

[ScopeTrigger Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerDigital Method

Configures the common properties of a digital trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax**C#**

```
public void ConfigureTriggerDigital(
    ScopeTriggerSource triggerSource,
    ScopeTriggerSlope triggerSlope,
    PrecisionTimeSpan triggerHoldoff,
    PrecisionTimeSpan triggerDelay
)
```

```
Public Sub ConfigureTriggerDigital (
    triggerSource As ScopeTriggerSource,
    triggerSlope As ScopeTriggerSlope,
```

```

triggerHoldoff As PrecisionTimeSpan,
triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerSlope

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSlope](#)

A value indicating whether you want a rising edge or a falling edge to trigger the digitizer.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerHysteresis Method

Configures common properties for analog hysteresis triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public void ConfigureTriggerHysteresis(
    ScopeTriggerSource triggerSource,
    double triggerLevel,
    double triggerhysteresisSize,
    ScopeTriggerSlope triggerSlope,
    ScopeTriggerCoupling triggerCoupling,
    PrecisionTimeSpan triggerHoldoff,
    PrecisionTimeSpan triggerDelay
)
```

```
Public Sub ConfigureTriggerHysteresis (
    triggerSource As ScopeTriggerSource,
    triggerLevel As Double,
    triggerhysteresisSize As Double,
    triggerSlope As ScopeTriggerSlope,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerLevel

Type: [SystemDouble](#)

The voltage threshold for the trigger.

triggerhysteresisSize

Type: [SystemDouble](#)

The size of the hysteresis window, in volts, on either side of the [Level](#).

triggerSlope

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSlope](#)

A value indicating whether you want a rising edge or a falling edge to trigger the digitizer.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstruments.PrecisionTimeSpan](#)

The time duration, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerImmediate Method

Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in [NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public void ConfigureTriggerImmediate()
```

```
Public Sub ConfigureTriggerImmediate
```

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerSoftware Method

Configures common properties for software triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public void ConfigureTriggerSoftware(  
    PrecisionTimeSpan triggerHoldoff,  
    PrecisionTimeSpan triggerDelay  
)
```

```
Public Sub ConfigureTriggerSoftware (  
    triggerHoldoff As PrecisionTimeSpan,  
    triggerDelay As PrecisionTimeSpan  
)
```

Parameters

triggerHoldoff

Type: NationalInstrumentsPrecisionTimeSpan

The length of time, in seconds, the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: `NationalInstrumentsPrecisionTimeSpan`

The length of time, in seconds, the digitizer waits after receiving the trigger to start acquiring data.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerConfigureTriggerWindow Method

Configures common properties for analog window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public void ConfigureTriggerWindow(  
    ScopeTriggerSource triggerSource,  
    double triggerLowLevel,  
    double triggerHighLevel,  
    ScopeWindowTriggerMode triggerWindowMode,  
    ScopeTriggerCoupling triggerCoupling,  
    PrecisionTimeSpan triggerHoldoff,  
    PrecisionTimeSpan triggerDelay  
)
```

```
Public Sub ConfigureTriggerWindow (  
    triggerSource As ScopeTriggerSource,  
    triggerLowLevel As Double,  
    triggerHighLevel As Double,  
    triggerWindowMode As ScopeWindowTriggerMode,
```

```

    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source.

triggerLowLevel

Type: [SystemDouble](#)

The voltage threshold the digitizer to uses for low triggering.

triggerHighLevel

Type: [SystemDouble](#)

The voltage threshold the digitizer to uses for high triggering.

triggerWindowMode

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWindowTriggerMode](#)

A value indicating whether the trigger should occur when the signal enters or leaves a window.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

See Also

Reference

[ScopeTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerCoupling Enumeration

Specifies how the digitizer couples the trigger source.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTriggerCoupling
```

```
Public Enumeration ScopeTriggerCoupling
```

Members

Member name	Value	Description
AC	0	The digitizer AC couples the input signal.
DC	1	The digitizer DC couples the input signal.
HighFrequencyReject	3	The digitizer filters the signal to remove high frequencies.
LowFrequencyReject	4	The digitizer filters the signal to remove low frequencies.
ACPlusHighFrequencyReject	1001	The digitizer AC couples the signal to remove high frequencies. The result is a passband.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerModifier Enumeration

Configures the device to automatically complete an acquisition if a trigger has not been received, after a period of time.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTriggerModifier
```

```
Public Enumeration ScopeTriggerModifier
```

Members

Member name	Value	Description
None	1	The acquisition is not triggered if the trigger Type is set to any value other than Immediate .
Auto	2	The acquisition is auto-triggered, if no trigger arrives.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerPolarity Enumeration

Specifies the polarity with which the digitizer will trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTriggerPolarity
```

```
Public Enumeration ScopeTriggerPolarity
```

Members

Member name	Value	Description
Positive	1	The digitizer will trigger with positive polarity.
Negative	2	The digitizer will trigger with negative polarity.
Either	3	The digitizer will trigger with either polarity.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSlope Enumeration

Specifies whether a rising or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTriggerSlope
```

```
Public Enumeration ScopeTriggerSlope
```

Members

Member name	Value	Description
Positive	1	A positive (rising) edge passing through the trigger level triggers the digitizer. For a digital trigger, this corresponds to a low-to-high transition.
Negative	0	A negative (falling) edge passing through the trigger level triggers the digitizer. For a digital trigger, this corresponds to a high-to-low transition.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Class

Represents the source terminal for [Source](#).

Inheritance Hierarchy

[SystemObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax































[C#VB](#)

```
public sealed class ScopeTriggerSource
```

```
Public NotInheritable Class ScopeTriggerSource
```

The ScopeTriggerSource type exposes the following members.








Properties

	Name	Description
	<u>Aux0Pfi0</u>	Gets the source terminal for AUX I/O connector 0 PFI 0.
	<u>Aux0Pfi1</u>	Gets the source terminal for AUX I/O connector 0 PFI 1.
	<u>Aux0Pfi2</u>	Gets the source terminal for AUX I/O connector 0 PFI 2.
	<u>Aux0Pfi3</u>	Gets the source terminal for AUX I/O connector 0 PFI 3.
	<u>Aux0Pfi4</u>	Gets the source terminal for AUX I/O connector 0 PFI 4.
	<u>Aux0Pfi5</u>	Gets the source terminal for AUX I/O connector 0 PFI 5.
	<u>Aux0Pfi6</u>	Gets the source terminal for AUX I/O connector 0 PFI 6.
	<u>Aux0Pfi7</u>	Gets the source terminal for AUX I/O connector 0 PFI 7.
	<u>Channel0</u>	Gets the source terminal for channel 0.
	<u>Channel1</u>	Gets the source terminal for channel 1.
	<u>Channel2</u>	Gets the source terminal for channel 2.
	<u>Channel3</u>	Gets the source terminal for channel 3.
	<u>Channel4</u>	Gets the source terminal for channel 4.
	<u>Channel5</u>	Gets the source terminal for channel 5.
	<u>Channel6</u>	Gets the source terminal for channel 6.
	<u>Channel7</u>	Gets the source terminal for channel 7.
	<u>External</u>	Gets the source terminal for External.
	<u>Immediate</u>	Gets the source terminal for Immediate.
	<u>Pfi0</u>	Gets the source terminal for PFI 0.
	<u>Pfi1</u>	Gets the source terminal for PFI 1.
	<u>Pfi2</u>	Gets the source terminal for PFI 2.
	<u>PxiStar</u>	Gets the source terminal for PXI STAR.
	<u>Rtsi0</u>	Gets the source terminal for RTSI 0.
	<u>Rtsi1</u>	Gets the source terminal for RTSI 1.
	<u>Rtsi2</u>	Gets the source terminal for RTSI 2.
	<u>Rtsi3</u>	Gets the source terminal for RTSI 3.
	<u>Rtsi4</u>	Gets the source terminal for RTSI 4.
	<u>Rtsi5</u>	Gets the source terminal for RTSI 5.
	<u>Rtsi6</u>	Gets the source terminal for RTSI 6.
	<u>Rtsi7</u>	Gets the source terminal for RTSI 7.

 SoftwareTriggerFunction	Gets the source terminal for the software trigger.
---	--









[Top](#)

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeTriggerSource)	Determines whether the current instance of ScopeTriggerSource and the specified ScopeTriggerSource object are equal.
 	FromString	Creates a ScopeTriggerSource object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeTriggerSource. (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeTriggerSource to String . (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
 	Equality	Checks whether two ScopeTriggerSource object instances are equal.
 	(String to ScopeTriggerSource)	Converts a specified String to an equivalent ScopeTriggerSource object.
 	(ScopeTriggerSource to String)	Converts a ScopeTriggerSource object to an equivalent String .
 	Inequality	Checks whether two ScopeTriggerSource object instances are not equal.

[Top](#)

Remarks

For more information see [Features Supported by Device](#) in the NI High-Speed Digitizers Help.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

[ScopeTriggerSource](#)











[NationalInstruments.ModularInstruments.NIScopeScopeExportedSignals](#)



















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Properties

The [ScopeTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	Aux0Pfi0	Gets the source terminal for AUX I/O connector 0 PFI 0.
	Aux0Pfi1	Gets the source terminal for AUX I/O connector 0 PFI 1.
	Aux0Pfi2	Gets the source terminal for AUX I/O connector 0 PFI 2.
	Aux0Pfi3	Gets the source terminal for AUX I/O connector 0 PFI 3.
	Aux0Pfi4	Gets the source terminal for AUX I/O connector 0 PFI 4.
	Aux0Pfi5	Gets the source terminal for AUX I/O connector 0 PFI 5.
	Aux0Pfi6	Gets the source terminal for AUX I/O connector 0 PFI 6.
	Aux0Pfi7	Gets the source terminal for AUX I/O connector 0 PFI 7.
	Channel0	Gets the source terminal for channel 0.
	Channel1	Gets the source terminal for channel 1.

 Channel2	Gets the source terminal for channel 2.
 Channel3	Gets the source terminal for channel 3.
 Channel4	Gets the source terminal for channel 4.
 Channel5	Gets the source terminal for channel 5.
 Channel6	Gets the source terminal for channel 6.
 Channel7	Gets the source terminal for channel 7.
 External	Gets the source terminal for External.
 Immediate	Gets the source terminal for Immediate.
 Pfi0	Gets the source terminal for PFI 0.
 Pfi1	Gets the source terminal for PFI 1.
 Pfi2	Gets the source terminal for PFI 2.
 PxiStar	Gets the source terminal for PXI STAR.
 Rtsi0	Gets the source terminal for RTSI 0.
 Rtsi1	Gets the source terminal for RTSI 1.
 Rtsi2	Gets the source terminal for RTSI 2.
 Rtsi3	Gets the source terminal for RTSI 3.
 Rtsi4	Gets the source terminal for RTSI 4.
 Rtsi5	Gets the source terminal for RTSI 5.
 Rtsi6	Gets the source terminal for RTSI 6.
 Rtsi7	Gets the source terminal for RTSI 7.
 SoftwareTriggerFunction	Gets the source terminal for the software trigger.

[Top](#)

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi0 Property

Gets the source terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi0 Property

Gets the source terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi0 { get; }
```



```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi1 Property

Gets the source terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi2 Property

Gets the source terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi3 Property

Gets the source terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi4 Property

Gets the source terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi5 Property

Gets the source terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi6 Property

Gets the source terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi7 Property

Gets the source terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel0 Property

Gets the source terminal for channel 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel0 { get; }
```

```
Public Shared ReadOnly Property Channel0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel1 Property

Gets the source terminal for channel 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel1 { get; }
```

```
Public Shared ReadOnly Property Channel1 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel2 Property

Gets the source terminal for channel 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel2 { get; }
```

```
Public Shared ReadOnly Property Channel2 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel3 Property

Gets the source terminal for channel 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel3 { get; }
```

```
Public Shared ReadOnly Property Channel3 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel4 Property

Gets the source terminal for channel 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel4 { get; }
```

```
Public Shared ReadOnly Property Channel4 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "4".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel5 Property

Gets the source terminal for channel 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel5 { get; }
```

```
Public Shared ReadOnly Property Channel5 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel6 Property

Gets the source terminal for channel 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel6 { get; }
```

```
Public Shared ReadOnly Property Channel6 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel7 Property

Gets the source terminal for channel 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel7 { get; }
```

```
Public Shared ReadOnly Property Channel7 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceExternal Property

Gets the source terminal for External.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource External { get; }
```

```
Public Shared ReadOnly Property External As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_EXTERNAL".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceSoftwareTriggerFunction Property

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeTriggerSource Class](#)































[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Properties

The [ScopeTriggerSource](#) type exposes the following members.

Properties

	Name	Description
	<u>Aux0Pfi0</u>	Gets the source terminal for AUX I/O connector 0 PFI 0.
	<u>Aux0Pfi1</u>	Gets the source terminal for AUX I/O connector 0 PFI 1.
	<u>Aux0Pfi2</u>	Gets the source terminal for AUX I/O connector 0 PFI 2.
	<u>Aux0Pfi3</u>	Gets the source terminal for AUX I/O connector 0 PFI 3.
	<u>Aux0Pfi4</u>	Gets the source terminal for AUX I/O connector 0 PFI 4.
	<u>Aux0Pfi5</u>	Gets the source terminal for AUX I/O connector 0 PFI 5.
	<u>Aux0Pfi6</u>	Gets the source terminal for AUX I/O connector 0 PFI 6.
	<u>Aux0Pfi7</u>	Gets the source terminal for AUX I/O connector 0 PFI 7.
	<u>Channel0</u>	Gets the source terminal for channel 0.
	<u>Channel1</u>	Gets the source terminal for channel 1.
	<u>Channel2</u>	Gets the source terminal for channel 2.
	<u>Channel3</u>	Gets the source terminal for channel 3.
	<u>Channel4</u>	Gets the source terminal for channel 4.
	<u>Channel5</u>	Gets the source terminal for channel 5.
	<u>Channel6</u>	Gets the source terminal for channel 6.
	<u>Channel7</u>	Gets the source terminal for channel 7.
	<u>External</u>	Gets the source terminal for External.
	<u>Immediate</u>	Gets the source terminal for Immediate.
	<u>Pfi0</u>	Gets the source terminal for PFI 0.
	<u>Pfi1</u>	Gets the source terminal for PFI 1.
	<u>Pfi2</u>	Gets the source terminal for PFI 2.
	<u>PxiStar</u>	Gets the source terminal for PXI STAR.
	<u>Rtsi0</u>	Gets the source terminal for RTSI 0.
	<u>Rtsi1</u>	Gets the source terminal for RTSI 1.
	<u>Rtsi2</u>	Gets the source terminal for RTSI 2.
	<u>Rtsi3</u>	Gets the source terminal for RTSI 3.
	<u>Rtsi4</u>	Gets the source terminal for RTSI 4.
	<u>Rtsi5</u>	Gets the source terminal for RTSI 5.
	<u>Rtsi6</u>	Gets the source terminal for RTSI 6.
	<u>Rtsi7</u>	Gets the source terminal for RTSI 7.

[SoftwareTriggerFunction](#)

Gets the source terminal for the software trigger.

[Top](#)[See Also](#)[Reference](#)[ScopeTriggerSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi0 Property

Gets the source terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

[Syntax](#)[C#VB](#)

```
public static ScopeTriggerSource Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeTriggerSource  
    Get
```

[Property Value](#)Type: [ScopeTriggerSource](#)A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_0".[See Also](#)[Reference](#)[ScopeTriggerSource Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi0 Property

Gets the source terminal for AUX I/O connector 0 PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi0 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi1 Property

Gets the source terminal for AUX I/O connector 0 PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi1 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi1 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi2 Property

Gets the source terminal for AUX I/O connector 0 PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi2 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi2 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi3 Property

Gets the source terminal for AUX I/O connector 0 PFI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi3 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi3 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi4 Property

Gets the source terminal for AUX I/O connector 0 PFI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi4 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi4 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_4".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi5 Property

Gets the source terminal for AUX I/O connector 0 PFI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi5 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi5 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi6 Property

Gets the source terminal for AUX I/O connector 0 PFI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi6 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi6 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceAux0Pfi7 Property

Gets the source terminal for AUX I/O connector 0 PFI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Aux0Pfi7 { get; }
```

```
Public Shared ReadOnly Property Aux0Pfi7 As ScopeTriggerSource  
    Get
```


Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_AUX_0_PFI_7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel0 Property

Gets the source terminal for channel 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel0 { get; }
```

```
Public Shared ReadOnly Property Channel0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "0".

See Also

Reference

[ScopeTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel1 Property

Gets the source terminal for channel 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel1 { get; }
```

```
Public Shared ReadOnly Property Channel1 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel2 Property

Gets the source terminal for channel 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel2 { get; }
```

```
Public Shared ReadOnly Property Channel2 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel3 Property

Gets the source terminal for channel 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel3 { get; }
```

```
Public Shared ReadOnly Property Channel3 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel4 Property

Gets the source terminal for channel 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel4 { get; }
```

```
Public Shared ReadOnly Property Channel4 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "4".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel5 Property

Gets the source terminal for channel 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel5 { get; }
```

```
Public Shared ReadOnly Property Channel5 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel6 Property

Gets the source terminal for channel 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel6 { get; }
```

```
Public Shared ReadOnly Property Channel6 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceChannel7 Property

Gets the source terminal for channel 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Channel7 { get; }
```

```
Public Shared ReadOnly Property Channel7 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceExternal Property

Gets the source terminal for External.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource External { get; }
```

```
Public Shared ReadOnly Property External As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_EXTERNAL".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceImmediate Property

Gets the source terminal for Immediate.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Immediate { get; }
```

```
Public Shared ReadOnly Property Immediate As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_IMMEDIATE".

See Also

Reference

[ScopeTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi0 Property

Gets the source terminal for PFI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi0 { get; }
```

```
Public Shared ReadOnly Property Pfi0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi1 Property

Gets the source terminal for PFI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi1 { get; }
```

```
Public Shared ReadOnly Property Pfi1 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePfi2 Property

Gets the source terminal for PFI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Pfi2 { get; }
```

```
Public Shared ReadOnly Property Pfi2 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PFI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourcePxiStar Property

Gets the source terminal for PXI STAR.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource PxiStar { get; }
```

```
Public Shared ReadOnly Property PxiStar As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_PXI_STAR".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi0 Property

Gets the source terminal for RTSI 0.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi0 { get; }
```

```
Public Shared ReadOnly Property Rtsi0 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_0".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi1 Property

Gets the source terminal for RTSI 1.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi1 { get; }
```

```
Public Shared ReadOnly Property Rtsi1 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_1".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi2 Property

Gets the source terminal for RTSI 2.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi2 { get; }
```

```
Public Shared ReadOnly Property Rtsi2 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_2".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi3 Property

Gets the source terminal for RTSI 3.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi3 { get; }
```

```
Public Shared ReadOnly Property Rtsi3 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_3".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi4 Property

Gets the source terminal for RTSI 4.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi4 { get; }
```

```
Public Shared ReadOnly Property Rtsi4 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_4".

See Also

Reference

[ScopeTriggerSource Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi5 Property

Gets the source terminal for RTSI 5.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi5 { get; }
```

```
Public Shared ReadOnly Property Rtsi5 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_5".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi6 Property

Gets the source terminal for RTSI 6.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi6 { get; }
```

```
Public Shared ReadOnly Property Rtsi6 As ScopeTriggerSource  
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTISI_6".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceRtsi7 Property

Gets the source terminal for RTSI 7.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource Rtsi7 { get; }
```

```
Public Shared ReadOnly Property Rtsi7 As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_RTSI_7".

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceSoftwareTriggerFunction Property

Gets the source terminal for the software trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource SoftwareTriggerFunction { get; }
```

```
Public Shared ReadOnly Property SoftwareTriggerFunction As ScopeTriggerSource
    Get
```

Property Value

Type: [ScopeTriggerSource](#)

A [ScopeTriggerSource](#) representing the [String](#) "VAL_SW_TRIG_FUNC".

See Also

Reference

[ScopeTriggerSource Class](#)








[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Methods

The [ScopeTriggerSource](#) type exposes the following members.

Methods

	Name	Description
	Equals(Object)	Determines whether the current instance of ScopeTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
	Equals(ScopeTriggerSource)	Determines whether the current instance of ScopeTriggerSource and the specified ScopeTriggerSource object are equal.
	 FromString	Creates a ScopeTriggerSource object from the specified String .
	GetHashCode	Gets the hash code for the current instance of ScopeTriggerSource . (Overrides ObjectGetHashCode .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Converts the current instance of ScopeTriggerSource to String . (Overrides ObjectToString .)

[Top](#)

See Also

Reference



[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeTriggerSource and the specified object are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeTriggerSource)	Determines whether the current instance of ScopeTriggerSource and the specified ScopeTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (ScopeTriggerSource)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified [ScopeTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the [ScopeTriggerSource](#) object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeTriggerSource and the specified object are equal. (Overrides Object.Equals(Object) .)
 Equals(ScopeTriggerSource)	Determines whether the current instance of ScopeTriggerSource and the specified ScopeTriggerSource object are equal.

[Top](#)

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(
    Object obj
)
```



```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (Object)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

Specifies the object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquals Method (ScopeTriggerSource)

Determines whether the current instance of [ScopeTriggerSource](#) and the specified [ScopeTriggerSource](#) object are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeTriggerSource source  
)
```

```
Public Function Equals (  
    source As ScopeTriggerSource  
) As Boolean
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the [ScopeTriggerSource](#) object to compare to the current instance of [ScopeTriggerSource](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceFromString Method

Creates a [ScopeTriggerSource](#) object from the specified [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static ScopeTriggerSource FromString(  
    string source  
)
```

```
Public Shared Function FromString (  
    source As String  
) As ScopeTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) that the [ScopeTriggerSource](#) object returned represents.

Return Value

Type: [ScopeTriggerSource](#)

Returns an object of type [ScopeTriggerSource](#) representing the specified source [String](#).

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource.GetHashCode Method

Gets the hash code for the current instance of [ScopeTriggerSource](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

Returns an [Int32](#) representing the hash code for the current instance of [ScopeTriggerSource](#).

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceToString Method

Converts the current instance of [ScopeTriggerSource](#) to [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

Returns a [String](#) representing the current instance of [ScopeTriggerSource](#).

See Also

Reference

[ScopeTriggerSource Class](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Operators and Type Conversions

The [ScopeTriggerSource](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether two ScopeTriggerSource object instances are equal.
	(String to ScopeTriggerSource)	Converts a specified String to an equivalent ScopeTriggerSource object.
	(ScopeTriggerSource to String)	Converts a ScopeTriggerSource object to an equivalent String .
	Inequality	Checks whether two ScopeTriggerSource object instances are not equal.

[Top](#)

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquality Operator

Checks whether two [ScopeTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeTriggerSource source1,
    ScopeTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeTriggerSource,
    source2 As ScopeTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the first [ScopeTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the second [ScopeTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceEquality Operator

Checks whether two [ScopeTriggerSource](#) object instances are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeTriggerSource source1,
    ScopeTriggerSource source2
)
```

```
Public Shared Operator = (
    source1 As ScopeTriggerSource,
    source2 As ScopeTriggerSource
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the first [ScopeTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the second [ScopeTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference



[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Conversion Operators

Overload List

	Name	Description
	(String to ScopeTriggerSource)	Converts a specified String to an equivalent ScopeTriggerSource object.
	(ScopeTriggerSource to String)	Converts a ScopeTriggerSource object to an equivalent String .

[Top](#)

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Conversion (String to ScopeTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeTriggerSource](#) object.

Return Value

Type: [ScopeTriggerSource](#)

Returns the [ScopeTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Conversion (String to ScopeTriggerSource)

Converts a specified [String](#) to an equivalent [ScopeTriggerSource](#) object.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator ScopeTriggerSource (  
    string source  
)
```

```
Public Shared Widening Operator CType (  
    source As String  
) As ScopeTriggerSource
```

Parameters

source

Type: [SystemString](#)

Specifies the [String](#) to be converted to an equivalent [ScopeTriggerSource](#) object.

Return Value

Type: [ScopeTriggerSource](#)

Returns the [ScopeTriggerSource](#) object representing the specified [String](#).

See Also

Reference

[ScopeTriggerSource Class](#)

[Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSource Conversion (ScopeTriggerSource to String)

Converts a [ScopeTriggerSource](#) object to an equivalent [String](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static implicit operator string (  
    ScopeTriggerSource source  
)
```

```
Public Shared Widening Operator CType (  
    source As ScopeTriggerSource  
) As String
```

Parameters

source

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the [ScopeTriggerSource](#) object to be converted to [String](#).

Return Value

Type: [String](#)

Returns the [String](#) representing the [ScopeTriggerSource](#) object.

See Also

Reference

[ScopeTriggerSource Class](#)
[Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerSourceInequality Operator

Checks whether two [ScopeTriggerSource](#) object instances are not equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator !=(  
    ScopeTriggerSource source1,  
    ScopeTriggerSource source2  
)
```

```
Public Shared Operator <> (  
    source1 As ScopeTriggerSource,  
    source2 As ScopeTriggerSource  
) As Boolean
```

Parameters

source1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the first [ScopeTriggerSource](#) object to compare.

source2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
Specifies the second [ScopeTriggerSource](#) object to compare.

Return Value

Type: [Boolean](#)

if the two instances are not equal; otherwise, .

See Also

Reference

[ScopeTriggerSource Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTriggerType Enumeration

Specifies the type of trigger to use.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public enum ScopeTriggerType
```

```
Public Enumeration ScopeTriggerType
```

Members

Member name	Value	Description
Edge	1	The digitizer is configured for edge triggering.
Width	2	The digitizer is configured for glitch triggering.
Runt	3	The digitizer is configured for width triggering.
Glitch	4	The digitizer is configured for runt triggering.
TV	5	The digitizer is configured for video triggering.
Immediate	6	The digitizer does not wait for a trigger of any kind.
Hysteresis	1001	The digitizer is configured for hysteresis triggering.
DigitalEdge	1002	The digitizer is configured for digital triggering.
Window	1003	The digitizer is configured for window triggering.
Software	1004	The digitizer waits until you call SendSoftwareEdgeTrigger .

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTrigger Class

Represents the [ScopeMeasurement](#) sub-object that contains properties and methods related to the video reference trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeTVTrigger

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB






```
public sealed class ScopeTVTrigger : ScopeSubObject,
    IIVI_ScopeTriggerTV
```

```
Public NotInheritable Class ScopeTVTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTriggerTV
```

The ScopeTVTrigger type exposes the following members.






Properties

Name	Description
------	-------------

 DCRestoreEnabled	Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.
 LineNumber	Gets or sets the line on which to trigger, if TriggerEvent is set to LineNumber .
 Polarity	Gets or sets whether the video signal sync is positive or negative.
 SignalFormat	Gets or sets the type of video signal.
 TriggerEvent	Gets or sets the condition in the video signal that causes the digitizer to trigger.

[Top](#)

Methods

Name	Description
 Configure	Configures the common properties for video triggering, including the signal format, TV event, line number, polarity, and enable DC restore. A video trigger occurs when the digitizer finds a valid video signal sync. When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the Source attribute. The default is Immediate . Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a method such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger method, you specify configuration parameters such as the trigger source and the amount of trigger delay.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger Video/TV attributes topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTrigger Properties

The [ScopeTVTrigger](#) type exposes the following members.

Properties

Name	Description
 DCRestoreEnabled	Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.
 LineNumber	Gets or sets the line on which to trigger, if TriggerEvent is set to LineNumber .
 Polarity	Gets or sets whether the video signal sync is positive or negative.
 SignalFormat	Gets or sets the type of video signal.
 TriggerEvent	Gets or sets the condition in the video signal that causes the digitizer to trigger.

[Top](#)

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerDCRestoreEnabled Property

Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DCRestoreEnabled { get; set; }
```

```
Public Property DCRestoreEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if DC restore is enabled and if DC restore is disabled.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerDCRestoreEnabled Property

Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DCRestoreEnabled { get; set; }
```

```
Public Property DCRestoreEnabled As Boolean  
    Get  
    Set
```

Property Value

Type: [Boolean](#)

if DC restore is enabled and if DC restore is disabled.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerLineNumber Property

Gets or sets the line on which to trigger, if [TriggerEvent](#) is set to [LineNumber](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long LineNumber { get; set; }
```

```
Public Property LineNumber As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The valid range of the property depends on the [SignalFormat](#) selected.

Signal Format	Line Numbers
M-NTSC, 480i, 480p	1 to 525
BG/PAL, SECAM, 576i, 576p	1 to 625
720p	1 to 750
1080i, 1080p	1 to 1125

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerPolarity Property

Gets or sets whether the video signal sync is positive or negative.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerPolarity Polarity { get; set; }
```

```
Public Property Polarity As ScopeTVTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTVTriggerPolarity](#)

Valid values for this property are [Negative](#) and [Positive](#).

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerSignalFormat Property

Gets or sets the type of video signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerSignalFormat SignalFormat { get; set; }
```

```
Public Property SignalFormat As ScopeTVTriggerSignalFormat
    Get
    Set
```

Property Value

Type: [ScopeTVTriggerSignalFormat](#)

Specifies the [ScopeTVTriggerSignalFormat](#).

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerTriggerEvent Property

Gets or sets the condition in the video signal that causes the digitizer to trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerEvent TriggerEvent { get; set; }
```

```
Public Property TriggerEvent As ScopeTVTriggerEvent  
    Get  
    Set
```

Property Value

Type: [ScopeTVTriggerEvent](#)

Specifies the [ScopeTVTriggerEvent](#).

See Also

Reference

[ScopeTVTrigger Class](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTrigger Properties

The [ScopeTVTrigger](#) type exposes the following members.

Properties

Name	Description
 DCRestoreEnabled	Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.
 LineNumber	Gets or sets the line on which to trigger, if TriggerEvent is set to LineNumber .
 Polarity	Gets or sets whether the video signal sync is positive or negative.
 SignalFormat	Gets or sets the type of video signal.
 TriggerEvent	Gets or sets the condition in the video signal that causes the digitizer to trigger.

[Top](#)

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerDCRestoreEnabled Property

Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DCRestoreEnabled { get; set; }
```

```
Public Property DCRestoreEnabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if DC restore is enabled and if DC restore is disabled.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerDCRestoreEnabled Property

Gets or sets whether the video-triggered data retrieved by the digitizer should be restored to the video signal's zero reference point.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool DCRestoreEnabled { get; set; }
```



```
Public Property DCRestoreEnabled As Boolean
    Get
    Set
```

Property Value

Type: [Boolean](#)

if DC restore is enabled and if DC restore is disabled.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerLineNumber Property

Gets or sets the line on which to trigger, if [TriggerEvent](#) is set to [LineNumber](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long LineNumber { get; set; }
```

```
Public Property LineNumber As Long
    Get
    Set
```

Property Value

Type: [Int64](#)

The valid range of the property depends on the [SignalFormat](#) selected.

Signal Format	Line Numbers
M-NTSC, 480i, 480p	1 to 525
BG/PAL, SECAM, 576i, 576p	1 to 625
720p	1 to 750
1080i, 1080p	1 to 1125

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerPolarity Property

Gets or sets whether the video signal sync is positive or negative.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerPolarity Polarity { get; set; }
```

```
Public Property Polarity As ScopeTVTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTVTriggerPolarity](#)

Valid values for this property are [Negative](#) and [Positive](#).

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerSignalFormat Property

Gets or sets the type of video signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerSignalFormat SignalFormat { get; set; }
```

```
Public Property SignalFormat As ScopeTVTriggerSignalFormat  
    Get  
    Set
```

Property Value

Type: [ScopeTVTriggerSignalFormat](#)

Specifies the [ScopeTVTriggerSignalFormat](#).

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerTriggerEvent Property

Gets or sets the condition in the video signal that causes the digitizer to trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeTVTriggerEvent TriggerEvent { get; set; }
```

```
Public Property TriggerEvent As ScopeTVTriggerEvent
    Get
    Set
```

Property Value

Type: [ScopeTVTriggerEvent](#)

Specifies the [ScopeTVTriggerEvent](#).

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)






Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTrigger Methods

The [ScopeTVTrigger](#) type exposes the following members.

Methods

Name	Description
------	-------------

 Configure	Configures the common properties for video triggering, including the signal format, TV event, line number, polarity, and enable DC restore. A video trigger occurs when the digitizer finds a valid video signal sync. When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the Source attribute. The default is Immediate . Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a method such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger method, you specify configuration parameters such as the trigger source and the amount of trigger delay.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerConfigure Method

Configures the common properties for video triggering, including the signal format, TV event, line number, polarity, and enable DC restore. A video trigger occurs when the digitizer finds a valid video signal sync. When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [Source](#) attribute. The default is [Immediate](#). Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points,

the digitizer waits for a reference (stop) trigger that you specify with a method such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger method, you specify configuration parameters such as the trigger source and the amount of trigger delay.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    ScopeTriggerSource triggerSource,
    bool triggerDCRestoreEnable,
    ScopeTVTriggerSignalFormat triggerSignalFormat,
    ScopeTVTriggerEvent triggerEvent,
    long triggerLineNumber,
    ScopeTVTriggerPolarity triggerPolarity,
    ScopeTriggerCoupling triggerCoupling,
    PrecisionTimeSpan triggerHoldoff,
    PrecisionTimeSpan triggerDelay
)
```

```
Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    triggerDCRestoreEnable As Boolean,
    triggerSignalFormat As ScopeTVTriggerSignalFormat,
    triggerEvent As ScopeTVTriggerEvent,
    triggerLineNumber As Long,
    triggerPolarity As ScopeTVTriggerPolarity,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source. Refer to [ScopeTriggerSource](#) for defined values.

triggerDCRestoreEnable

Type: [SystemBoolean](#)

Offsets each video line so the clamping level (the portion of the video line between the end of the color burst and the beginning of the active image) is moved to zero volt. Refer to [DCRestoreEnabled](#) for defined values.

triggerSignalFormat

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerSignalFormat](#)

The type of video signal sync the digitizer should look for. Refer to [SignalFormat](#) for more information.

triggerEvent

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerEvent](#)

The TV event you want to trigger on. You can trigger on a specific or on the next coming line or field of the signal.

triggerLineNumber

Type: [SystemInt64](#)

The line number to trigger on. The line number range covers an entire frame and is referenced as shown on Vertical Blanking and Synchronization Signal. Refer to [LineNumber](#) for more information. Default value is 1

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerPolarity](#)

The polarity of the video signal sync.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

Applies coupling and filtering options to the trigger signal. Refer to [Coupling](#) for more information.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to [Holdoff](#) for more information.

triggerDelay

Type: `NationalInstrumentsPrecisionTimeSpan`

The amount of time the digitizer waits after receiving the trigger to start acquiring data. Refer to [Delay](#) for more information.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerConfigure Method

Configures the common properties for video triggering, including the signal format, TV event, line number, polarity, and enable DC restore. A video trigger occurs when the digitizer finds a valid video signal sync. When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the [Source](#) attribute. The default is [Immediate](#). Upon receiving the start trigger the digitizer begins sampling pretrigger points. After the digitizer finishes sampling pretrigger points, the digitizer waits for a reference (stop) trigger that you specify with a method such as this one. Upon receiving the reference trigger the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger method, you specify configuration parameters such as the trigger source and the amount of trigger delay.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: `NationalInstruments.ModularInstruments.NIScope.Fx45` (in `NationalInstruments.ModularInstruments.NIScope.Fx45.dll`) Version: 2020

Syntax

C#VB

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    bool triggerDCRestoreEnable,  
    ScopeTVTriggerSignalFormat triggerSignalFormat,
```



```

ScopeTVTriggerEvent triggerEvent,
long triggerLineNumber,
ScopeTVTriggerPolarity triggerPolarity,
ScopeTriggerCoupling triggerCoupling,
PrecisionTimeSpan triggerHoldoff,
PrecisionTimeSpan triggerDelay
)

```

```

Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    triggerDCRestoreEnable As Boolean,
    triggerSignalFormat As ScopeTVTriggerSignalFormat,
    triggerEvent As ScopeTVTriggerEvent,
    triggerLineNumber As Long,
    triggerPolarity As ScopeTVTriggerPolarity,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)

```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)

The trigger source. Refer to [ScopeTriggerSource](#) for defined values.

triggerDCRestoreEnable

Type: [SystemBoolean](#)

Offsets each video line so the clamping level (the portion of the video line between the end of the color burst and the beginning of the active image) is moved to zero volt. Refer to [DCRestoreEnabled](#) for defined values.

triggerSignalFormat

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerSignalFormat](#)

The type of video signal sync the digitizer should look for. Refer to [SignalFormat](#) for more information.

triggerEvent

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerEvent](#)

The TV event you want to trigger on. You can trigger on a specific or on the next coming line or field of the signal.

triggerLineNumber

Type: [SystemInt64](#)

The line number to trigger on. The line number range covers an entire frame and is referenced as shown on Vertical Blanking and Synchronization Signal.

Refer to [LineNumber](#) for more information. Default value is 1

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTVTriggerPolarity](#)

The polarity of the video signal sync.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

Applies coupling and filtering options to the trigger signal. Refer to [Coupling](#) for more information.

triggerHoldoff

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The length of time the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger. Refer to [Holdoff](#) for more information.

triggerDelay

Type: [NationalInstrumentsPrecisionTimeSpan](#)

The amount of time the digitizer waits after receiving the trigger to start acquiring data. Refer to [Delay](#) for more information.

See Also

Reference

[ScopeTVTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerEvent Enumeration

Specifies the condition in the video signal that causes the digitizer to trigger.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in

[NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax

C#VB

```
public enum ScopeTVTriggerEvent
```

```
Public Enumeration ScopeTVTriggerEvent
```

Members

Member name	Value	Description
Field1	1	A trigger on field 1 of the signal.
Field2	2	A trigger on field 2 of the signal.
AnyField	3	A trigger on the first field acquired.
AnyLine	4	A trigger on the first line acquired.
LineNumber	5	A trigger on a specific line of a video signal. Valid values vary depending on the SignalFormat configured.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerPolarity Enumeration

Specifies whether a rising edge or a falling edge triggers the digitizer.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTVTriggerPolarity
```

```
Public Enumeration ScopeTVTriggerPolarity
```

Members

Member name	Value	Description
Positive	1	A positive (rising) edge passing through the trigger level triggers the digitizer. For a digital trigger, this corresponds to a low-to-high transition.
Negative	2	A negative (falling) edge passing through the trigger level triggers the digitizer. For a digital trigger, this corresponds to a high-to-low transition.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeTVTriggerSignalFormat Enumeration

Specifies the type of video signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeTVTriggerSignalFormat
```

```
Public Enumeration ScopeTVTriggerSignalFormat
```

Members

Member name	Value	Description
-------------	-------	-------------

Ntsc	1	An NTSC signal format that supports line numbers from 1 to 525.
Pal	2	A PAL signal format that supports line numbers from 1 to 625.
Secam	3	A SECAM signal format that supports line numbers from 1 to 625.
MPal	1001	An M-PAL signal format that supports line numbers from 1 to 525.
I480Lines5994FieldsPerSecond	1010	A signal that contains 480 lines, interlaced, 59.94 fields per second.
I480Lines60FieldsPerSecond	1011	A signal that contains 480 lines, interlaced, 60 fields per second.
P480Lines5994FramesPerSecond	1015	A signal that contains 480 lines, progressive, 59.94 fields per second.
P480Lines60FramesPerSecond	1016	A signal that contains 480 lines, progressive, 60 frames per second.
I576Lines50FieldsPerSecond	1020	A signal that contains 576 lines, interlaced, 50 fields per second.
P576Lines50FramesPerSecond	1025	A signal that contains 576 lines, progressive, 50 frames per second.
P720Lines50FramesPerSecond	1031	A signal that contains 720 lines, progressive, 50 frames per second.
P720Lines5994FramesPerSecond	1032	A signal that contains 720 lines, progressive, 59.94 frames per second.
P720Lines60FramesPerSecond	1033	A signal that contains 720 lines, progressive, 60 frames per second.
I1080Lines50FieldsPerSecond	1040	A signal that contains 1,080 lines, interlaced, 50 fields per second.
I1080Lines5994FieldsPerSecond	1041	A signal that contains 1,080 lines, interlaced, 59.94 fields per second.
I1080Lines60FieldsPerSecond	1042	A signal that contains 1,080 lines, interlaced, 60 fields per second.
P1080Lines24FramesPerSecond	1045	A signal that contains 1,080 lines, progressive, 24 frames per second.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeVerticalCoupling Enumeration

Specifies how the digitizer couples the input signal for the channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeVerticalCoupling
```

```
Public Enumeration ScopeVerticalCoupling
```

Members

Member name	Value	Description
AC	0	AC vertical coupling.
DC	1	DC vertical coupling.
Ground	2	Ground vertical coupling.

Remarks

When input coupling changes, the input stage takes a finite amount of time to settle.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning Class

Provides warning codes for the warnings raised by the underlying driver.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeWarning

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax





C#VB












```
public sealed class ScopeWarning
```

```
Public NotInheritable Class ScopeWarning
```

The ScopeWarning type exposes the following members.






Properties


	Name	Description
	AdcOverload	Gets the warning code when one or more channels has an ADC overload.
	AutoSetupNoSignal	Gets the warning code when auto Setup was unable to find a signal.
	ChannelOverload	Gets the warning code when the 50 ohm protection on one or more channels has been disabled because of excess signal power.
	Code	Gets the GUID code assigned to the warning.

 DataOverflow	Gets the warning code when an Onboard Signal Processing feature has caused the input data to overflow outside the valid vertical range.
 ErrorQueryNotSupportedWarningCode	Gets the warning code when Error Query is not supported.
 HeaterCircuitTemperature	Gets the warning code when onboard temperature stabilization circuit requires calibration.
 InvalidData	Gets the warning code when data acquired during the acquisition is possibly invalid.
 InvalidWaveformElement	Gets the warning code when one of the elements in the waveform array is invalid.
 P2POverflow	Gets the warning code when one or more peer-to-peer endpoints has overflowed.
 PLLUnlocked	Gets the warning code when the PLL is not locked.
 PLLUnlockedAndAdcOverload	Gets the warning code when the PLL is not locked and the ADC has an overload.
 Text	Gets the message related to the warning.
 TimestampRollOver	Gets the warning code when the timestamp counter has rolled over.
 UnexpectedDriverWarningCode	Gets the warning code for an unexpected driver warning.

[Top](#)



Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeWarning and the object that you specify are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeWarning)	Determines whether the current instance of ScopeWarning and the ScopeWarning object that you specify are equal.
 Equals(ScopeWarning, Boolean)	Checks for the equality for the two ScopeWarning objects considering whether warning message should be compared or not.
 GetHashCode	Returns the hash code for the current instance of ScopeWarning. (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance.

		(Inherited from Object .)
	ToString	Converts the current instance of ScopeWarning to string. (Overrides ObjectToString .)

[Top](#)

Operators

	Name	Description
	Equality	Checks whether the two instances of ScopeWarning are equal.
	Inequality	Checks whether the two instances of ScopeWarning are unequal.

[Top](#)

Remarks

Represents the list of warnings returned by the driver.

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference


[NationalInstruments.ModularInstruments.NIScope Namespace](#)















Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning Properties

The [ScopeWarning](#) type exposes the following members.

Properties

	Name	Description
	AdcOverload	Gets the warning code when one or more channels has an ADC overload.

 AutoSetupNoSignal	Gets the warning code when auto Setup was unable to find a signal.
 ChannelOverload	Gets the warning code when the 50 ohm protection on one or more channels has been disabled because of excess signal power.
 Code	Gets the GUID code assigned to the warning.
 DataOverflow	Gets the warning code when an Onboard Signal Processing feature has caused the input data to overflow outside the valid vertical range.
 ErrorQueryNotSupportedWarningCode	Gets the warning code when Error Query is not supported.
 HeaterCircuitTemperature	Gets the warning code when onboard temperature stabilization circuit requires calibration.
 InvalidData	Gets the warning code when data acquired during the acquisition is possibly invalid.
 InvalidWaveformElement	Gets the warning code when one of the elements in the waveform array is invalid.
 P2POverflow	Gets the warning code when one or more peer-to-peer endpoints has overflowed.
 PLLUnlocked	Gets the warning code when the PLL is not locked.
 PLLUnlockedAndAdcOverload	Gets the warning code when the PLL is not locked and the ADC has an overload.
 Text	Gets the message related to the warning.
 TimestampRollOver	Gets the warning code when the timestamp counter has rolled over.
 UnexpectedDriverWarningCode	Gets the warning code for an unexpected driver warning.

[Top](#)

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAdcOverload Property

Gets the warning code when one or more channels has an ADC overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AdcOverload { get; }
```

```
Public Shared ReadOnly Property AdcOverload As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("12676361-21C7-4CF6-89FD-C02AB003F5FF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAdcOverload Property

Gets the warning code when one or more channels has an ADC overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AdcOverload { get; }
```

```
Public Shared ReadOnly Property AdcOverload As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("12676361-21C7-4CF6-89FD-C02AB003F5FF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAutoSetupNoSignal Property

Gets the warning code when auto Setup was unable to find a signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AutoSetupNoSignal { get; }
```

```
Public Shared ReadOnly Property AutoSetupNoSignal As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("FCB15664-9823-40D1-9457-AECABBB7BCA2") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningChannelOverload Property

Gets the warning code when the 50 ohm protection on one or more channels has been disabled because of excess signal power.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static Guid ChannelOverload { get; }
```

```
Public Shared ReadOnly Property ChannelOverload As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("3BC10F42-A518-4686-A2DA-B661EAF7C3E6") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningCode Property

Gets the GUID code assigned to the warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid  
    Get
```

Property Value

Type: [Guid](#)

The [Guid](#) code of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningDataOverflow Property

Gets the warning code when an Onboard Signal Processing feature has caused the input data to overflow outside the valid vertical range.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid DataOverflow { get; }
```

```
Public Shared ReadOnly Property DataOverflow As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("4AF26653-23DE-4B47-BFF2-1C0B11031E64") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningErrorQueryNotSupportedWarningCode

Gets the warning code when Error Query is not supported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid ErrorQueryNotSupportedWarningCode { get; }
```

```
Public Shared ReadOnly Property ErrorQueryNotSupportedWarningCode As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("BE37BF5D-FAE5-44d0-8AA4-4B521D1D17DE") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningHeaterCircuitTemperature Property

Gets the warning code when onboard temperature stabilization circuit requires calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid HeaterCircuitTemperature { get; }
```



```
Public Shared ReadOnly Property HeaterCircuitTemperature As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("FB1D49D6-CC01-4EC3-A546-59AD1DC217CF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningInvalidData Property

Gets the warning code when data acquired during the acquisition is possibly invalid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid InvalidData { get; }
```

```
Public Shared ReadOnly Property InvalidData As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("07263369-AA19-42A5-8F18-005CEB754A65") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningInvalidWaveformElement Property

Gets the warning code when one of the elements in the waveform array is invalid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid InvalidWaveformElement { get; }
```

```
Public Shared ReadOnly Property InvalidWaveformElement As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("2F1F7321-C82D-4CB4-B907-D70A5C31940B") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningP2POverflow Property

Gets the warning code when one or more peer-to-peer endpoints has overflowed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid P2POverflow { get; }
```

```
Public Shared ReadOnly Property P2POverflow As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("B04AC88D-10A0-447E-B483-6F140A8A8911") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningPllUnlocked Property

Gets the warning code when the PLL is not locked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid PllUnlocked { get; }
```

```
Public Shared ReadOnly Property PllUnlocked As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("8A45A6EC-28E0-4BED-B7FA-A7B2F1F11ED4") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningPllUnlockedAndAdcOverload Property

Gets the warning code when the PLL is not locked and the ADC has an overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid PllUnlockedAndAdcOverload { get; }
```

```
Public Shared ReadOnly Property PllUnlockedAndAdcOverload As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("4283A5F6-C8D8-4107-8D43-C70DC1A3991A") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningText Property

Gets the message related to the warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

The warning message.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningTimestampRollOver Property

Gets the warning code when the timestamp counter has rolled over.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid TimestampRollOver { get; }
```

```
Public Shared ReadOnly Property TimestampRollOver As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("5ED8BC0C-C2D2-4683-87D7-14E078A945A2") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningUnexpectedDriverWarningCode Property

Gets the warning code for an unexpected driver warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid UnexpectedDriverWarningCode { get; }
```

```
Public Shared ReadOnly Property UnexpectedDriverWarningCode As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("2cca167a-dbe8-4ba4-88dd-8b8faaf7b119") of the warning.

See Also

Reference

[ScopeWarning Class](#)
















[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning Properties

The [ScopeWarning](#) type exposes the following members.

Properties

	Name	Description
	<u>AdcOverload</u>	Gets the warning code when one or more channels has an ADC overload.
	<u>AutoSetupNoSignal</u>	Gets the warning code when auto Setup was unable to find a signal.
	<u>ChannelOverload</u>	Gets the warning code when the 50 ohm protection on one or more channels has been disabled because of excess signal power.
	<u>Code</u>	Gets the GUID code assigned to the warning.
	<u>DataOverflow</u>	Gets the warning code when an Onboard Signal Processing feature has caused the input data to overflow outside the valid vertical range.
	<u>ErrorQueryNotSupportedWarningCode</u>	Gets the warning code when Error Query is not supported.
	<u>HeaterCircuitTemperature</u>	Gets the warning code when onboard temperature stabilization circuit requires calibration.
	<u>InvalidData</u>	Gets the warning code when data acquired during the acquisition is possibly invalid.
	<u>InvalidWaveformElement</u>	Gets the warning code when one of the elements in the waveform array is invalid.
	<u>P2POverflow</u>	Gets the warning code when one or more peer-to-peer endpoints has overflowed.
	<u>PLLUnlocked</u>	Gets the warning code when the PLL is not locked.
	<u>PLLUnlockedAndAdcOverload</u>	Gets the warning code when the PLL is not locked and the ADC has an overload.
	<u>Text</u>	Gets the message related to the warning.
	<u>TimestampRollOver</u>	Gets the warning code when the timestamp counter has rolled over.
	<u>UnexpectedDriverWarningCode</u>	Gets the warning code for an unexpected driver warning.

[Top](#)

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAdcOverload Property

Gets the warning code when one or more channels has an ADC overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AdcOverload { get; }
```

```
Public Shared ReadOnly Property AdcOverload As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("12676361-21C7-4CF6-89FD-C02AB003F5FF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAdcOverload Property

Gets the warning code when one or more channels has an ADC overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AdcOverload { get; }
```

```
Public Shared ReadOnly Property AdcOverload As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("12676361-21C7-4CF6-89FD-C02AB003F5FF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningAutoSetupNoSignal Property

Gets the warning code when auto Setup was unable to find a signal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid AutoSetupNoSignal { get; }
```

```
Public Shared ReadOnly Property AutoSetupNoSignal As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("FCB15664-9823-40D1-9457-AECABBB7BCA2") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningChannelOverload Property

Gets the warning code when the 50 ohm protection on one or more channels has been disabled because of excess signal power.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid ChannelOverload { get; }
```

```
Public Shared ReadOnly Property ChannelOverload As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("3BC10F42-A518-4686-A2DA-B661EAF7C3E6") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningCode Property

Gets the GUID code assigned to the warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid  
    Get
```

Property Value

Type: [Guid](#)

The [Guid](#) code of the warning.

See Also

Reference

[ScopeWarning Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningDataOverflow Property

Gets the warning code when an Onboard Signal Processing feature has caused the input data to overflow outside the valid vertical range.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid DataOverflow { get; }
```

```
Public Shared ReadOnly Property DataOverflow As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("4AF26653-23DE-4B47-BFF2-1C0B11031E64") of the warning.

See Also

Reference

ScopeWarning Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningErrorQueryNotSupportedWarningCode Property

Gets the warning code when Error Query is not supported.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid ErrorQueryNotSupportedWarningCode { get; }
```

```
Public Shared ReadOnly Property ErrorQueryNotSupportedWarningCode As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("BE37BF5D-FAE5-44d0-8AA4-4B521D1D17DE") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningHeaterCircuitTemperature Property

Gets the warning code when onboard temperature stabilization circuit requires calibration.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid HeaterCircuitTemperature { get; }
```

```
Public Shared ReadOnly Property HeaterCircuitTemperature As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("FB1D49D6-CC01-4EC3-A546-59AD1DC217CF") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningInvalidData Property

Gets the warning code when data acquired during the acquisition is possibly invalid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid InvalidData { get; }
```

```
Public Shared ReadOnly Property InvalidData As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("07263369-AA19-42A5-8F18-005CEB754A65") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningInvalidWaveformElement Property

Gets the warning code when one of the elements in the waveform array is invalid.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public static Guid InvalidWaveformElement { get; }
```

```
Public Shared ReadOnly Property InvalidWaveformElement As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("2F1F7321-C82D-4CB4-B907-D70A5C31940B") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningP2POverflow Property

Gets the warning code when one or more peer-to-peer endpoints has overflowed.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid P2POverflow { get; }
```

```
Public Shared ReadOnly Property P2POverflow As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("B04AC88D-10A0-447E-B483-6F140A8A8911") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningPllUnlocked Property

Gets the warning code when the PLL is not locked.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid PllUnlocked { get; }
```

```
Public Shared ReadOnly Property PllUnlocked As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("8A45A6EC-28E0-4BED-B7FA-A7B2F1F11ED4") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningPllUnlockedAndAdcOverload Property

Gets the warning code when the PLL is not locked and the ADC has an overload.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid PllUnlockedAndAdcOverload { get; }
```

```
Public Shared ReadOnly Property PllUnlockedAndAdcOverload As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("4283A5F6-C8D8-4107-8D43-C70DC1A3991A") of the warning.

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningText Property

Gets the message related to the warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String  
    Get
```

Property Value

Type: [String](#)

The warning message.

See Also

Reference

[ScopeWarning Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningTimestampRollOver Property

Gets the warning code when the timestamp counter has rolled over.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid TimestampRollOver { get; }
```

```
Public Shared ReadOnly Property TimestampRollOver As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID ("5ED8BC0C-C2D2-4683-87D7-14E078A945A2") of the warning.

See Also

Reference

ScopeWarning Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningUnexpectedDriverWarningCode Property

Gets the warning code for an unexpected driver warning.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static Guid UnexpectedDriverWarningCode { get; }
```

```
Public Shared ReadOnly Property UnexpectedDriverWarningCode As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID ("2cca167a-dbe8-4ba4-88dd-8b8faaf7b119") of the warning.

See Also

Reference

[ScopeWarning Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)





Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning Methods

The [ScopeWarning](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeWarning and the object that you specify are equal. (Overrides Object.Equals(Object) .)
 Equals(ScopeWarning)	Determines whether the current instance of ScopeWarning and the ScopeWarning object that you specify are equal.

 Equals(ScopeWarning, Boolean)	Checks for the equality for the two ScopeWarning objects considering whether warning message should be compared or not.
 GetHashCode	Returns the hash code for the current instance of ScopeWarning . (Overrides ObjectGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Converts the current instance of ScopeWarning to string. (Overrides ObjectToString .)

[Top](#)

See Also




Reference

[ScopeWarning Class](#)[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeWarning and the object that you specify are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeWarning)	Determines whether the current instance of ScopeWarning and the ScopeWarning object that you specify are equal.
 Equals(ScopeWarning, Boolean)	Checks for the equality for the two ScopeWarning objects considering whether warning message should be compared or not.

[Top](#)

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (Object)

Determines whether the current instance of [ScopeWarning](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (Object)

Determines whether the current instance of [ScopeWarning](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (ScopeWarning)

Determines whether the current instance of [ScopeWarning](#) and the [ScopeWarning](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeWarning warning  
)
```

```
Public Function Equals (  
    warning As ScopeWarning  
) As Boolean
```

Parameters

warning

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
The [ScopeWarning](#) object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (ScopeWarning, Boolean)

Checks for the equality for the two [ScopeWarning](#) objects considering whether warning message should be compared or not.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeWarning warning,
```

```
bool ignoreWarningMessage
)
```

```
Public Function Equals (
    warning As ScopeWarning,
    ignoreWarningMessage As Boolean
) As Boolean
```

Parameters

warning

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
The [ScopeWarning](#) object to which this object is compared.

ignoreWarningMessage

Type: [SystemBoolean](#)

Specifies if the warning message must be ignored. If the warning message is to be ignored, then value of ignoreWarningMessage is . If warning message is to be compared then value of ignoreWarningMessage is .

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)




[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method

Overload List

Name	Description
 Equals(Object)	Determines whether the current instance of ScopeWarning and the object that you specify are equal. (Overrides ObjectEquals(Object) .)
 Equals(ScopeWarning)	Determines whether the current instance of ScopeWarning and the ScopeWarning object that you specify are equal.
 Equals(ScopeWarning, Boolean)	Checks for the equality for the two ScopeWarning objects considering whether warning message should be compared or not.

[Top](#)

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (Object)

Determines whether the current instance of [ScopeWarning](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (Object)

Determines whether the current instance of [ScopeWarning](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (ScopeWarning)

Determines whether the current instance of [ScopeWarning](#) and the [ScopeWarning](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeWarning warning  
)
```

```
Public Function Equals (  
    warning As ScopeWarning  
) As Boolean
```

Parameters

warning

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
The [ScopeWarning](#) object to be compared to the current instance of [ScopeWarning](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquals Method (ScopeWarning, Boolean)

Checks for the equality for the two [ScopeWarning](#) objects considering whether warning message should be compared or not.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(
    ScopeWarning warning,
    bool ignoreWarningMessage
)
```

```
Public Function Equals (
    warning As ScopeWarning,
    ignoreWarningMessage As Boolean
) As Boolean
```

Parameters

warning

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
The [ScopeWarning](#) object to which this object is compared.

ignoreWarningMessage

Type: [SystemBoolean](#)

Specifies if the warning message must be ignored. If the warning message is to be ignored, then value of ignoreWarningMessage is . If warning message is to be compared then value of ignoreWarningMessage is .

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[Equals Overload](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning.GetHashCode Method

Returns the hash code for the current instance of [ScopeWarning](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) that represents the hash code for the current instance of [ScopeWarning](#).

See Also

Reference

ScopeWarning Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning.ToString Method

Converts the current instance of [ScopeWarning](#) to string.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

Return Value

Type: [String](#)

A [String](#) that represents the current instance of [ScopeWarning](#).

See Also

Reference

[ScopeWarning Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarning Operators

The [ScopeWarning](#) type exposes the following members.

Operators

	Name	Description
	Equality	Checks whether the two instances of ScopeWarning are equal.
	Inequality	Checks whether the two instances of ScopeWarning are unequal.

[Top](#)

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquality Operator

Checks whether the two instances of [ScopeWarning](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeWarning warning1,
    ScopeWarning warning2
)
```

```
Public Shared Operator = (
    warning1 As ScopeWarning,
    warning2 As ScopeWarning
) As Boolean
```

Parameters

warning1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object.

warning2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object for the comparison.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEquality Operator

Checks whether the two instances of [ScopeWarning](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeWarning warning1,
    ScopeWarning warning2
)
```

```
Public Shared Operator = (
    warning1 As ScopeWarning,
    warning2 As ScopeWarning
) As Boolean
```

Parameters

warning1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object.

warning2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object for the comparison.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningInequality Operator

Checks whether the two instances of [ScopeWarning](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static bool operator !=(
    ScopeWarning warning1,
    ScopeWarning warning2
)
```

```
Public Shared Operator <> (
    warning1 As ScopeWarning,
    warning2 As ScopeWarning
) As Boolean
```

Parameters

warning1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object.

warning2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWarning](#)
A [ScopeWarning](#) object.

Return Value

Type: [Boolean](#)

if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeWarning Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgs Class

Represents the properties under the Warning Event Args raised in case of a driver warning.

Inheritance Hierarchy

[SystemObject](#)

[SystemEventArgs](#)

[NationalInstruments.ModularInstruments.NIScopeScopeWarningEventArgs](#)

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: [NationalInstruments.ModularInstruments.NIScope.Fx45](#) (in

[NationalInstruments.ModularInstruments.NIScope.Fx45.dll](#)) Version: 2020

Syntax




C#VB

```
public sealed class ScopeWarningEventArgs : EventArgs
```

```
Public NotInheritable Class ScopeWarningEventArgs
    Inherits EventArgs
```





The `ScopeWarningEventArgs` type exposes the following members.

Properties

	Name	Description
	Code	Gets the Warning Event Args code.
	Text	Gets the Warning Event Args message.
	Warning	Gets the warning set in the Warning Event Args.

[Top](#)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgs Properties

The [ScopeWarningEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the Warning Event Args code.
	Text	Gets the Warning Event Args message.
	Warning	Gets the warning set in the Warning Event Args.

[Top](#)

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsCode Property

Gets the Warning Event Args code.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID code of the warning.

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsCode Property

Gets the Warning Event Args code.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)
Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID code of the warning.

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsText Property

Gets the Warning Event Args message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the message.

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgs.Warning Property

Gets the warning set in the Warning Event Args.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWarning Warning { get; }
```

```
Public ReadOnly Property Warning As ScopeWarning  
    Get
```

Property Value

Type: [ScopeWarning](#)

An object of type [ScopeWarning](#).

See Also

Reference

[ScopeWarningEventArgs Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgs Properties

The [ScopeWarningEventArgs](#) type exposes the following members.

Properties

	Name	Description
	Code	Gets the Warning Event Args code.
	Text	Gets the Warning Event Args message.
	Warning	Gets the warning set in the Warning Event Args.

[Top](#)

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsCode Property

Gets the Warning Event Args code.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid
    Get
```

Property Value

Type: [Guid](#)

The GUID code of the warning.

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsCode Property

Gets the Warning Event Args code.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public Guid Code { get; }
```

```
Public ReadOnly Property Code As Guid  
    Get
```

Property Value

Type: [Guid](#)

The GUID code of the warning.

See Also

Reference

[ScopeWarningEventArgs Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsText Property

Gets the Warning Event Args message.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public string Text { get; }
```

```
Public ReadOnly Property Text As String  
    Get
```

Property Value

Type: [String](#)

A [String](#) representing the message.

See Also

Reference

ScopeWarningEventArgs Class

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgsWarning Property

Gets the warning set in the Warning Event Args.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWarning Warning { get; }
```

```
Public ReadOnly Property Warning As ScopeWarning
    Get
```

Property Value

Type: [ScopeWarning](#)

An object of type [ScopeWarning](#).

See Also

Reference

[ScopeWarningEventArgs Class](#)




[NationalInstruments.ModularInstruments.NIScope Namespace](#)


Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWarningEventArgs Methods

The [ScopeWarningEventArgs](#) type exposes the following members.

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance.

	(Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeWarningEventArgs Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfo Structure

Contains timing and scaling information about the waveform obtained from the Fetch and Read methods.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB






```
public struct ScopeWaveformInfo
```

```
Public Structure ScopeWaveformInfo
```

The ScopeWaveformInfo type exposes the following members.






Properties

Name	Description
 AbsoluteInitialX	Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.

 ActualNumberOfSamples	Gets the actual number of samples fetched and placed in the waveform array.
 Gain	Gets the gain factor of the given channel.
 Offset	Gets the offset factor of the given channel.
 RelativeInitialX	Gets the time, in seconds, from the trigger to the first sample in the fetched waveform.
 XIncrement	Gets the time, in seconds, between points in the acquired waveform.



[Top](#)

Methods

Name	Description
 Equals(Object)	Determines if the current instance of ScopeWaveformInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeWaveformInfo)	Determines whether the current instance of ScopeWaveformInfo and the ScopeWaveformInfo object that you specify are equal.
 GetHashCode	Returns the hash code for the current instance of ScopeWaveformInfo. (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

Operators

Name	Description
 Equality	Checks whether the two instances of ScopeWaveformInfo are equal.
 Inequality	Checks whether the two instances of ScopeWaveformInfo are unequal.

[Top](#)

Remarks

For more information, refer to [NI High-Speed Digitizers Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfo Properties

The [ScopeWaveformInfo](#) type exposes the following members.

Properties

Name	Description
 AbsoluteInitialX	Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.
 ActualNumberOfSamples	Gets the actual number of samples fetched and placed in the waveform array.
 Gain	Gets the gain factor of the given channel.
 Offset	Gets the offset factor of the given channel.
 RelativeInitialX	Gets the time, in seconds, from the trigger to the first sample in the fetched waveform.
 XIncrement	Gets the time, in seconds, between points in the acquired waveform.

[Top](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoAbsoluteInitialX Property

Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double AbsoluteInitialX { get; }
```

```
Public ReadOnly Property AbsoluteInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Timestamp, in seconds, of the first fetched sample

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoAbsoluteInitialX Property

Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double AbsoluteInitialX { get; }
```

```
Public ReadOnly Property AbsoluteInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Timestamp, in seconds, of the first fetched sample

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoActualNumberOfSamples Property

Gets the actual number of samples fetched and placed in the waveform array.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long ActualNumberOfSamples { get; }
```

```
Public ReadOnly Property ActualNumberOfSamples As Long
    Get
```

Property Value

Type: [Int64](#)

Actual number of samples fetched and placed in the waveform array.

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double
    Get
```

Property Value

Type: [Double](#)

Gain factor of the given channel.

Remarks

This property is useful for scaling binary data with this formula:
voltage = binary data * gain factor + [Offset](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoOffset Property

Gets the offset factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Offset { get; }
```

```
Public ReadOnly Property Offset As Double  
    Get
```

Property Value

Type: [Double](#)

Offset factor of the given channel.

Remarks

This property useful for scaling binary data with this formula:
voltage = binary data * [Gain](#) + Offset

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoRelativeInitialX Property

Gets the time, in seconds, from the trigger to the first sample in the fetched waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RelativeInitialX { get; }
```

```
Public ReadOnly Property RelativeInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Time, in seconds, from the trigger to the first sample in the fetched waveform.

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoXIncrement Property

Gets the time, in seconds, between points in the acquired waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double XIncrement { get; }
```

```
Public ReadOnly Property XIncrement As Double  
    Get
```

Property Value

Type: [Double](#)

Time, in seconds, between points in the acquired waveform.

See Also

Reference

[ScopeWaveformInfo Structure](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfo Properties

The [ScopeWaveformInfo](#) type exposes the following members.

Properties

Name	Description
 AbsoluteInitialX	Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.
 ActualNumberOfSamples	Gets the actual number of samples fetched and placed in the waveform array.
 Gain	Gets the gain factor of the given channel.
 Offset	Gets the offset factor of the given channel.
 RelativeInitialX	Gets the time, in seconds, from the trigger to the first sample in the fetched waveform.
 XIncrement	Gets the time, in seconds, between points in the acquired waveform.

[Top](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoAbsoluteInitialX Property

Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in

NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double AbsoluteInitialX { get; }
```

```
Public ReadOnly Property AbsoluteInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Timestamp, in seconds, of the first fetched sample

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoAbsoluteInitialX Property

Gets the timestamp, in seconds, of the first fetched sample that is comparable between records and acquisitions; all digitizers do not support this field.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double AbsoluteInitialX { get; }
```

```
Public ReadOnly Property AbsoluteInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Timestamp, in seconds, of the first fetched sample

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoActualNumberOfSamples Property

Gets the actual number of samples fetched and placed in the waveform array.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public long ActualNumberOfSamples { get; }
```

```
Public ReadOnly Property ActualNumberOfSamples As Long  
    Get
```

Property Value

Type: [Int64](#)

Actual number of samples fetched and placed in the waveform array.

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoGain Property

Gets the gain factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Gain { get; }
```

```
Public ReadOnly Property Gain As Double  
    Get
```

Property Value

Type: [Double](#)

Gain factor of the given channel.

Remarks

This property is useful for scaling binary data with this formula:
 $\text{voltage} = \text{binary data} * \text{gain factor} + \text{Offset}$

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoOffset Property

Gets the offset factor of the given channel.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double Offset { get; }
```

```
Public ReadOnly Property Offset As Double  
    Get
```

Property Value

Type: [Double](#)

Offset factor of the given channel.

Remarks

This property useful for scaling binary data with this formula:
 $\text{voltage} = \text{binary data} * \text{Gain} + \text{Offset}$

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoRelativeInitialX Property

Gets the time, in seconds, from the trigger to the first sample in the fetched waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double RelativeInitialX { get; }
```

```
Public ReadOnly Property RelativeInitialX As Double  
    Get
```

Property Value

Type: [Double](#)

Time, in seconds, from the trigger to the first sample in the fetched waveform.

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoXIncrement Property

Gets the time, in seconds, between points in the acquired waveform.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double XIncrement { get; }
```

```
Public ReadOnly Property XIncrement As Double
    Get
```

Property Value

Type: [Double](#)

Time, in seconds, between points in the acquired waveform.

See Also

Reference

[ScopeWaveformInfo Structure](#)






[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfo Methods

The [ScopeWaveformInfo](#) type exposes the following members.

Methods

Name	Description
 Equals(Object)	Determines if the current instance of ScopeWaveformInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeWaveformInfo)	Determines whether the current instance of ScopeWaveformInfo and the ScopeWaveformInfo object that you specify are equal.
 GetHashCode	Returns the hash code for the current instance of ScopeWaveformInfo . (Overrides ValueTypeGetHashCode .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

[Top](#)

See Also

Reference



[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method

Overload List

Name	Description
 Equals(Object)	Determines if the current instance of ScopeWaveformInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeWaveformInfo)	Determines whether the current instance of ScopeWaveformInfo and the ScopeWaveformInfo object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (Object)

Determines if the current instance of [ScopeWaveformInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (Object)

Determines if the current instance of [ScopeWaveformInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (ScopeWaveformInfo)

Determines whether the current instance of [ScopeWaveformInfo](#) and the [ScopeWaveformInfo](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeWaveformInfo waveformInfo  
)
```

```
Public Function Equals (  
    waveformInfo As ScopeWaveformInfo  
) As Boolean
```

Parameters

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
The [ScopeWaveformInfo](#) object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)



[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method

Overload List

Name	Description
 Equals(Object)	Determines if the current instance of ScopeWaveformInfo and the object that you specify are equal. (Overrides ValueTypeEquals(Object) .)
 Equals(ScopeWaveformInfo)	Determines whether the current instance of ScopeWaveformInfo and the ScopeWaveformInfo object that you specify are equal.

[Top](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (Object)

Determines if the current instance of [ScopeWaveformInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (Object)

Determines if the current instance of [ScopeWaveformInfo](#) and the object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

Parameters

obj

Type: [SystemObject](#)

The object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquals Method (ScopeWaveformInfo)

Determines whether the current instance of [ScopeWaveformInfo](#) and the [ScopeWaveformInfo](#) object that you specify are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public bool Equals(  
    ScopeWaveformInfo waveformInfo  
)
```

```
Public Function Equals (  
    waveformInfo As ScopeWaveformInfo  
) As Boolean
```

Parameters

waveformInfo

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
The [ScopeWaveformInfo](#) object to be compared to the current instance of [ScopeWaveformInfo](#).

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[Equals Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoGetHashCode Method

Returns the hash code for the current instance of [ScopeWaveformInfo](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

Return Value

Type: [Int32](#)

An [Int32](#) that represents the hash code for the current instance of [ScopeWaveformInfo](#).

See Also

Reference

[ScopeWaveformInfo Structure](#)





[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfo Operators

The [ScopeWaveformInfo](#) type exposes the following members.

Operators

	Name	Description
 	Equality	Checks whether the two instances of ScopeWaveformInfo are equal.
 	Inequality	Checks whether the two instances of ScopeWaveformInfo are unequal.

[Top](#)

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquality Operator

Checks whether the two instances of [ScopeWaveformInfo](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)[VB](#)

```
public static bool operator ==(
    ScopeWaveformInfo waveformInfo1,
    ScopeWaveformInfo waveformInfo2
)
```

```
Public Shared Operator = (
    waveformInfo1 As ScopeWaveformInfo,
    waveformInfo2 As ScopeWaveformInfo
) As Boolean
```

Parameters

waveformInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

waveformInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoEquality Operator

Checks whether the two instances of [ScopeWaveformInfo](#) are equal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public static bool operator ==(
    ScopeWaveformInfo waveformInfo1,
    ScopeWaveformInfo waveformInfo2
)
```

```
Public Shared Operator = (  
    waveformInfo1 As ScopeWaveformInfo,  
    waveformInfo2 As ScopeWaveformInfo  
) As Boolean
```

Parameters

waveformInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

waveformInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

Return Value

Type: [Boolean](#)

if the two instances are equal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWaveformInfoInequality Operator

Checks whether the two instances of [ScopeWaveformInfo](#) are unequal.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in
NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)
[VB](#)

```
public static bool operator !=(
    ScopeWaveformInfo waveformInfo1,
    ScopeWaveformInfo waveformInfo2
)
```

```
Public Shared Operator <> (
    waveformInfo1 As ScopeWaveformInfo,
    waveformInfo2 As ScopeWaveformInfo
) As Boolean
```

Parameters

waveformInfo1

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

waveformInfo2

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWaveformInfo](#)
A [ScopeWaveformInfo](#) object.

Return Value

Type: [Boolean](#)

if the two instances are unequal; otherwise, .

See Also

Reference

[ScopeWaveformInfo Structure](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTrigger Class

Represents the sub-object that contains methods and properties used to configure common properties for width triggering.

Inheritance Hierarchy

SystemObject

NationalInstruments.ModularInstruments.NIScopeScopeSubObject
 NationalInstruments.ModularInstruments.NIScopeScopeWidthTrigger

Namespace: NationalInstruments.ModularInstruments.NIScope

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax





C#VB

```
public sealed class ScopeWidthTrigger : ScopeSubObject,
    IIVI_ScopeTriggerWidth
```

```
Public NotInheritable Class ScopeWidthTrigger
    Inherits ScopeSubObject
    Implements IIVI_ScopeTriggerWidth
```

The ScopeWidthTrigger type exposes the following members.

Properties

Name	Description
 ThresholdHigh	Gets or sets the upper time threshold you want the digitizer to use for width triggering.
 ThresholdLow	Gets or sets the lower time threshold you want the digitizer to use for width triggering.
 WidthCondition	Gets or sets the width condition you want the digitizer to use for width triggering.
 WidthPolarity	Gets or sets the width polarity you want the digitizer to use for width triggering.

Top

Methods

Name	Description
------	-------------

⇒ Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)	Configures common properties for analog width triggering.
⇒ Configure(ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog width triggering.
⇒ Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
⇒ GetHashCode	Serves as the default hash function. (Inherited from Object .)
⇒ GetType	Gets the Type of the current instance. (Inherited from Object .)
⇒ ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger related topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTrigger Properties

The [ScopeWidthTrigger](#) type exposes the following members.

Properties

Name	Description
 ThresholdHigh	Gets or sets the upper time threshold you want the digitizer to use for width triggering.
 ThresholdLow	Gets or sets the lower time threshold you want the digitizer to use for width triggering.
 WidthCondition	Gets or sets the width condition you want the digitizer to use for width triggering.
 WidthPolarity	Gets or sets the width polarity you want the digitizer to use for width triggering.

[Top](#)

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdHigh Property

Gets or sets the upper time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdLow](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdHigh Property

Gets or sets the upper time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdLow](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

NationalInstruments.ModularInstruments.NIScope Namespace

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdLow Property

Gets or sets the lower time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan ThresholdLow { get; set; }
```

```
Public Property ThresholdLow As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The lower voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdLow was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdHigh](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerWidthCondition Property

Gets or sets the width condition you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWidthTriggerCondition WidthCondition { get; set; }
```

```
Public Property WidthCondition As ScopeWidthTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeWidthTriggerCondition](#)

The width condition you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
-----------	-----------

ObjectDisposedException	WidthCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for width condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width is within or outside of the threshold window.

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerWidthPolarity Property

Gets or sets the width polarity you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)

```
public ScopeTriggerPolarity WidthPolarity { get; set; }
```

```
Public Property WidthPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The width polarity you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	WidthPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for width polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width is positive, negative or either.

See Also

Reference

[ScopeWidthTrigger Class](#)



[NationalInstruments.ModularInstruments.NIScope Namespace](#)



Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTrigger Properties

The [ScopeWidthTrigger](#) type exposes the following members.

Properties

Name	Description
 ThresholdHigh	Gets or sets the upper time threshold you want the digitizer to use for width triggering.
 ThresholdLow	Gets or sets the lower time threshold you want the digitizer to use for width triggering.

 WidthCondition	Gets or sets the width condition you want the digitizer to use for width triggering.
 WidthPolarity	Gets or sets the width polarity you want the digitizer to use for width triggering.

[Top](#)

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdHigh Property

Gets or sets the upper time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public PrecisionTimeSpan ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for upper threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdLow](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdHigh Property

Gets or sets the upper time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan ThresholdHigh { get; set; }
```

```
Public Property ThresholdHigh As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The upper voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdHigh was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for upper threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdLow](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerThresholdLow Property

Gets or sets the lower time threshold you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public PrecisionTimeSpan ThresholdLow { get; set; }
```

```
Public Property ThresholdLow As PrecisionTimeSpan
    Get
    Set
```

Property Value

Type: PrecisionTimeSpan

The lower voltage threshold you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	ThresholdLow was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for lower threshold was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the trigger signal enters or leaves the threshold window specified with this property and [ThresholdHigh](#).

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerWidthCondition Property

Gets or sets the width condition you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWidthTriggerCondition WidthCondition { get; set; }
```

```
Public Property WidthCondition As ScopeWidthTriggerCondition
    Get
    Set
```

Property Value

Type: [ScopeWidthTriggerCondition](#)

The width condition you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	WidthCondition was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for width condition was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width is within or outside of the threshold window.

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerWidthPolarity Property

Gets or sets the width polarity you want the digitizer to use for width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#](#)/[VB](#)

```
public ScopeTriggerPolarity WidthPolarity { get; set; }
```

```
Public Property WidthPolarity As ScopeTriggerPolarity
    Get
    Set
```

Property Value

Type: [ScopeTriggerPolarity](#)

The width polarity you want the digitizer to use for width triggering.

Exceptions

Exception	Condition
ObjectDisposedException	WidthPolarity was accessed after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	The value for width polarity was invalid. - or - The underlying NI-SCOPE driver returned an error.

Remarks

The digitizer triggers when the width is positive, negative or either.

See Also

Reference

[ScopeWidthTrigger Class](#)







[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTrigger Methods

The [ScopeWidthTrigger](#) type exposes the following members.

Methods

Name	Description
 Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)	Configures common properties for analog width triggering.
 Configure(ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog width triggering.
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference



[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method

Overload List

Name	Description
 Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)	Configures common properties for analog width triggering.
 Configure(ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog width triggering.

[Top](#)

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    string source,  
    double level,  
    PrecisionTimeSpan thresholdLow,  
    PrecisionTimeSpan thresholdHigh,  
    Polarity polarity,  
    WidthCondition condition  
)
```

```
Public Sub Configure (  
    source As String,  
    level As Double,  
    thresholdLow As PrecisionTimeSpan,  
    thresholdHigh As PrecisionTimeSpan,  
    polarity As Polarity,  
    condition As WidthCondition  
)
```

Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the pulse will trigger.

thresholdLow

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width high threshold.

thresholdHigh

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width low threshold.

polarity

Type: Ivi.ScopePolarity

The polarity that the width will trigger, either positive, negative, or either.

condition

Type: Ivi.ScopeWidthCondition

The condition for the pulse to trigger on width greater than or less than the threshold.

Implements

IviScopeTriggerWidthConfigure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for polarity was invalid. - or -</p> <p>The value for width condition was invalid. - or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public void Configure(  
    string source,  
    double level,  
    PrecisionTimeSpan thresholdLow,  
    PrecisionTimeSpan thresholdHigh,  
    Polarity polarity,  
    WidthCondition condition  
)
```

```
Public Sub Configure (  
    source As String,  
    level As Double,  
    thresholdLow As PrecisionTimeSpan,  
    thresholdHigh As PrecisionTimeSpan,  
    polarity As Polarity,  
    condition As WidthCondition  
)
```

Parameters

source

Type: [SystemString](#)

The trigger source.

levelType: [SystemDouble](#)

A value indicating the voltage level at which the pulse will trigger.

thresholdLowType: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width high threshold.

thresholdHighType: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width low threshold.

polarityType: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

conditionType: [Ivi.ScopeWidthCondition](#)

The condition for the pulse to trigger on width greater than or less than the threshold.

Implements[IviScopeTriggerWidthConfigure\(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition\)](#)**Exceptions**

Exception	Condition
ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid.</p> <p>- or -</p> <p>The value for upper threshold was invalid.</p> <p>- or -</p> <p>The value for polarity was invalid.</p> <p>- or -</p> <p>The value for width condition was invalid.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    ScopeTriggerSource triggerSource,  
    double level,  
    double triggerLowThreshold,  
    double triggerHighThreshold,  
    ScopeTriggerPolarity triggerPolarity,  
    ScopeWidthTriggerCondition triggerWidthCondition,  
    ScopeTriggerCoupling triggerCoupling,  
    PrecisionTimeSpan triggerHoldoff,
```

```
PrecisionTimeSpan triggerDelay
)
```

```
Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    level As Double,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As ScopeTriggerPolarity,
    triggerWidthCondition As ScopeWidthTriggerCondition,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
The trigger source.

level

Type: [SystemDouble](#)
A value indicating the voltage level at which the pulse will trigger.

triggerLowThreshold

Type: [SystemDouble](#)
A value indicating the pulse width low threshold.

triggerHighThreshold

Type: [SystemDouble](#)
A value indicating the pulse width high threshold.

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)
The polarity that the width will trigger, either positive, negative, or either.

triggerWidthCondition

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWidthTriggerCondition](#)
The condition for the pulse to trigger on width greater than or less than the threshold.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure(ScopeTriggerSource , Double, Double, Double, ScopeTriggerPolarity , ScopeWidthTriggerCondition , ScopeTriggerCoupling , PrecisionTimeSpan , PrecisionTimeSpan) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for polarity was invalid. - or -</p> <p>The value for width condition was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid.</p>

- or -

The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method

Overload List

Name	Description
Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)	Configures common properties for analog width triggering.
Configure(ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)	Configures common properties for analog width triggering.

[Top](#)

See Also

Reference

[ScopeWidthTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#/VB

```
public void Configure(  
    string source,  
    double level,  
    PrecisionTimeSpan thresholdLow,  
    PrecisionTimeSpan thresholdHigh,  
    Polarity polarity,  
    WidthCondition condition  
)
```

```
Public Sub Configure (  
    source As String,  
    level As Double,  
    thresholdLow As PrecisionTimeSpan,  
    thresholdHigh As PrecisionTimeSpan,  
    polarity As Polarity,  
    condition As WidthCondition  
)
```

Parameters

source

Type: [SystemString](#)

The trigger source.

levelType: [SystemDouble](#)

A value indicating the voltage level at which the pulse will trigger.

thresholdLowType: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width high threshold.

thresholdHighType: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width low threshold.

polarityType: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

conditionType: [Ivi.ScopeWidthCondition](#)

The condition for the pulse to trigger on width greater than or less than the threshold.

Implements

[IviScopeTriggerWidthConfigure\(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition\)](#)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid.</p> <p>- or -</p> <p>The value for upper threshold was invalid.</p> <p>- or -</p> <p>The value for polarity was invalid.</p> <p>- or -</p> <p>The value for width condition was invalid.</p> <p>- or -</p> <p>The underlying NI-SCOPE driver returned an error.</p>

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(  
    string source,  
    double level,  
    PrecisionTimeSpan thresholdLow,  
    PrecisionTimeSpan thresholdHigh,  
    Polarity polarity,  
    WidthCondition condition  
)
```

```
Public Sub Configure (  
    source As String,  
    level As Double,  
    thresholdLow As PrecisionTimeSpan,  
    thresholdHigh As PrecisionTimeSpan,  
    polarity As Polarity,  
    condition As WidthCondition  
)
```


Parameters

source

Type: [SystemString](#)

The trigger source.

level

Type: [SystemDouble](#)

A value indicating the voltage level at which the pulse will trigger.

thresholdLow

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width high threshold.

thresholdHigh

Type: [Ivi.DriverPrecisionTimeSpan](#)

A value indicating the pulse width low threshold.

polarity

Type: [Ivi.ScopePolarity](#)

The polarity that the width will trigger, either positive, negative, or either.

condition

Type: [Ivi.ScopeWidthCondition](#)

The condition for the pulse to trigger on width greater than or less than the threshold.

Implements

[IviScopeTriggerWidthConfigure\(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition\)](#)

Exceptions

Exception	Condition
ObjectDisposedException	Configure(String, Double, PrecisionTimeSpan, PrecisionTimeSpan, Polarity, WidthCondition) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviDriverException	The value for lower threshold was invalid. - or - The value for upper threshold was invalid. - or -

The value for polarity was invalid.
 - or -
 The value for width condition was invalid.
 - or -
 The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerConfigure Method (ScopeTriggerSource, Double, Double, Double, ScopeTriggerPolarity, ScopeWidthTriggerCondition, ScopeTriggerCoupling, PrecisionTimeSpan, PrecisionTimeSpan)

Configures common properties for analog width triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public void Configure(
    ScopeTriggerSource triggerSource,
    double level,
    double triggerLowThreshold,
    double triggerHighThreshold,
    ScopeTriggerPolarity triggerPolarity,
    ScopeWidthTriggerCondition triggerWidthCondition,
    ScopeTriggerCoupling triggerCoupling,
    PrecisionTimeSpan triggerHoldoff,
```

```
PrecisionTimeSpan triggerDelay
)
```

```
Public Sub Configure (
    triggerSource As ScopeTriggerSource,
    level As Double,
    triggerLowThreshold As Double,
    triggerHighThreshold As Double,
    triggerPolarity As ScopeTriggerPolarity,
    triggerWidthCondition As ScopeWidthTriggerCondition,
    triggerCoupling As ScopeTriggerCoupling,
    triggerHoldoff As PrecisionTimeSpan,
    triggerDelay As PrecisionTimeSpan
)
```

Parameters

triggerSource

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerSource](#)
The trigger source.

level

Type: [SystemDouble](#)
A value indicating the voltage level at which the pulse will trigger.

triggerLowThreshold

Type: [SystemDouble](#)
A value indicating the pulse width low threshold.

triggerHighThreshold

Type: [SystemDouble](#)
A value indicating the pulse width high threshold.

triggerPolarity

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerPolarity](#)
The polarity that the width will trigger, either positive, negative, or either.

triggerWidthCondition

Type: [NationalInstruments.ModularInstruments.NIScopeScopeWidthTriggerCondition](#)
The condition for the pulse to trigger on width greater than or less than the threshold.

triggerCoupling

Type: [NationalInstruments.ModularInstruments.NIScopeScopeTriggerCoupling](#)

A value indicating whether to apply coupling and filtering options to the trigger signal.

triggerHoldoff

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after detecting a trigger before enabling NI-SCOPE to detect another trigger.

triggerDelay

Type: [NationalInstruments.PrecisionTimeSpan](#)

The length of time, in seconds, that the digitizer waits after receiving the trigger to start acquiring data.

Exceptions

Exception	Condition
ArgumentNullException	ScopeTriggerSource object passed in was null.
ObjectDisposedException	Configure (ScopeTriggerSource , Double, Double, Double, ScopeTriggerPolarity , ScopeWidthTriggerCondition , ScopeTriggerCoupling , PrecisionTimeSpan , PrecisionTimeSpan) was called after the associated NIScope or ScopeDriverUtility object was disposed.
IviCDriverException	<p>The value for lower threshold was invalid. - or -</p> <p>The value for upper threshold was invalid. - or -</p> <p>The value for polarity was invalid. - or -</p> <p>The value for width condition was invalid. - or -</p> <p>The value for coupling was invalid. - or -</p> <p>The value for holdoff was invalid. - or -</p> <p>The value for delay was invalid.</p>

- or -

The underlying NI-SCOPE driver returned an error.

See Also

Reference

[ScopeWidthTrigger Class](#)

[Configure Overload](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWidthTriggerCondition Enumeration

Specifies whether a pulse that is within or outside the high and low threshold triggers the oscilloscope.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeWidthTriggerCondition
```

```
Public Enumeration ScopeWidthTriggerCondition
```

Members

Member name	Value	Description
Within	1	Width of pulse is within threshold.
Outside	2	Width of pulse is outside threshold.

Remarks

The end-user specifies the [ThresholdHigh](#) and [ThresholdLow](#)

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTrigger Class

Represents sub-objects of ScopeWindowReferenceTrigger that contain properties related to the window reference trigger.

Inheritance Hierarchy

[SystemObject](#)

[NationalInstruments.ModularInstruments.NIScopeScopeSubObject](#)

NationalInstruments.ModularInstruments.NIScopeScopeWindowReferenceTrigger

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax


C#VB



```
public sealed class ScopeWindowReferenceTrigger : ScopeSubObject
```

```
Public NotInheritable Class ScopeWindowReferenceTrigger
    Inherits ScopeSubObject
```

The ScopeWindowReferenceTrigger type exposes the following members.





Properties

Name	Description
 HighLevel	Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.

 LowLevel	Gets or sets the lower voltage threshold you want the digitizer to use for window triggering.
 Mode	Gets or sets whether a trigger must occur when the signal enters or leaves the window as specified by LowLevel or HighLevel .

[Top](#)

Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
 GetHashCode	Serves as the default hash function. (Inherited from Object .)
 GetType	Gets the Type of the current instance. (Inherited from Object .)
 ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

Remarks

For more information, refer to the Trigger Window attributes topics in the [NI-SCOPE Function Reference Help](#).

Thread Safety

Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources




[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTrigger Properties

The [ScopeWindowReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 HighLevel	Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.
 LowLevel	Gets or sets the lower voltage threshold you want the digitizer to use for window triggering.
 Mode	Gets or sets whether a trigger must occur when the signal enters or leaves the window as specified by LowLevel or HighLevel .

[Top](#)

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerHighLevel Property

Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

[C#VB](#)


```
public double HighLevel { get; set; }
```

```
Public Property HighLevel As Double
    Get
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

High Trigger Level \leq Vertical Range/2 + Vertical Offset

High Trigger Level \geq (-Vertical Range/2) + Vertical Offset

High Trigger Level > Low Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with HighLevel and this property.

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerHighLevel Property

Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighLevel { get; set; }
```

```
Public Property HighLevel As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

High Trigger Level \leq Vertical Range/2 + Vertical Offset

High Trigger Level \geq (-Vertical Range/2) + Vertical Offset

High Trigger Level > Low Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with HighLevel and this property.

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerLowLevel Property

Gets or sets the lower voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowLevel { get; set; }
```

```
Public Property LowLevel As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

Low Trigger Level \leq Vertical Range/2 + Vertical Offset

Low Trigger Level \geq (-Vertical Range/2) + Vertical Offset

Low Trigger Level < High Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with this property and [HighLevel](#).

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerMode Property

Gets or sets whether a trigger must occur when the signal enters or leaves the window as specified by [LowLevel](#) or [HighLevel](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWindowTriggerMode Mode { get; set; }
```

```
Public Property Mode As ScopeWindowTriggerMode  
    Get  
    Set
```

Property Value

Type: [ScopeWindowTriggerMode](#)

Specifies the [ScopeWindowTriggerMode](#).

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)




Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTrigger Properties

The [ScopeWindowReferenceTrigger](#) type exposes the following members.

Properties

Name	Description
 HighLevel	Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.
 LowLevel	Gets or sets the lower voltage threshold you want the digitizer to use for window triggering.
 Mode	Gets or sets whether a trigger must occur when the signal enters or leaves the window as specified by LowLevel or HighLevel .

[Top](#)

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)
[NationalInstruments.ModularInstruments.NIScope Namespace](#)
 Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerHighLevel Property

Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighLevel { get; set; }
```

```
Public Property HighLevel As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

High Trigger Level \leq Vertical Range/2 + Vertical Offset

High Trigger Level \geq (-Vertical Range/2) + Vertical Offset

High Trigger Level > Low Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with HighLevel and this property.

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerHighLevel Property

Gets or sets the upper voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double HighLevel { get; set; }
```

```
Public Property HighLevel As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

High Trigger Level \leq Vertical Range/2 + Vertical Offset

High Trigger Level \geq (-Vertical Range/2) + Vertical Offset

High Trigger Level > Low Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with HighLevel and this property.

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerLowLevel Property

Gets or sets the lower voltage threshold you want the digitizer to use for window triggering.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public double LowLevel { get; set; }
```

```
Public Property LowLevel As Double  
    Get  
    Set
```

Property Value

Type: [Double](#)

The values for this property must meet the following conditions:

Low Trigger Level \leq Vertical Range/2 + Vertical Offset

Low Trigger Level \geq (-Vertical Range/2) + Vertical Offset

Low Trigger Level < High Trigger Level

Remarks

The digitizer triggers when the trigger signal enters or leaves the window specified with this property and [HighLevel](#).

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources

[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTriggerMode Property

Gets or sets whether a trigger must occur when the signal enters or leaves the window as specified by [LowLevel](#) or [HighLevel](#).

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public ScopeWindowTriggerMode Mode { get; set; }
```

```
Public Property Mode As ScopeWindowTriggerMode
    Get
    Set
```

Property Value

Type: [ScopeWindowTriggerMode](#)
Specifies the [ScopeWindowTriggerMode](#).

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Other Resources





[Mapping the NI-SCOPE .NET API to the NI-SCOPE C API](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowReferenceTrigger Methods

The [ScopeWindowReferenceTrigger](#) type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object .)
	GetHashCode	Serves as the default hash function. (Inherited from Object .)
	GetType	Gets the Type of the current instance. (Inherited from Object .)
	ToString	Returns a string that represents the current object. (Inherited from Object .)

[Top](#)

See Also

Reference

[ScopeWindowReferenceTrigger Class](#)

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

ScopeWindowTriggerMode Enumeration

Specifies whether the trigger occurs when the signal enters or leaves a window.

Namespace: [NationalInstruments.ModularInstruments.NIScope](#)

Assembly: NationalInstruments.ModularInstruments.NIScope.Fx45 (in NationalInstruments.ModularInstruments.NIScope.Fx45.dll) Version: 2020

Syntax

C#VB

```
public enum ScopeWindowTriggerMode
```

```
Public Enumeration ScopeWindowTriggerMode
```

Members

Member name	Value	Description
Entering	0	Trigger occurs when the signal enters a window.
Leaving	1	Trigger occurs when the signal leaves a window.

See Also

Reference

[NationalInstruments.ModularInstruments.NIScope Namespace](#)

Copyright © National Instruments Corporation. All Rights Reserved.

Glossary



Prefixes

Symbol	Prefix	Value
p	pico	10^{-12}
n	nano	10^{-9}
μ	micro	10^{-6}
m	milli	10^{-3}
k	kilo	10^3
M	mega	10^6
G	giga	10^9
T	tera	10^{12}

A

AC	alternating current
AC-coupled	Allowing the transmission of AC signals while blocking DC signals.
AC coupling cutoff	The boundary between DC and AC signals.
accuracy	A measure of the capability of an instrument or sensor to faithfully indicate the value of the measured signal. This term is not related to resolution; however, the accuracy level can never be better than the resolution of the instrument.
active probe	An oscilloscope or instrument probe with built-in, active electronics. The built-in electronics are used to condition the signal—for example, to give a well-defined impedance or to connect the probe to a very high impedance circuit, whose signal would be degraded if transmitted over a longer signal lead. Active probes are especially used with high-frequency

	oscilloscopes, logic analyzers, and RF instruments.
ADE, application development environment	A software environment incorporating the development, debug, and analysis tools for software development. LabVIEW, Measurement Studio, and Visual Studio are examples.
<u>AGC</u>	automatic gain control—Applies gain to small signals to increase their bit resolution, so the signals can be processed by the DDC's resampling/interpolating halfband filters and output formatter.
amplitude	The voltage amplitude of a signal. When speaking of the amplitude of a signal, it is usually assumed the rms value for an AC signal. However, amplitude can also refer to the instantaneous amplitude, or the peak, peak-to-peak, or average amplitude, if so specified.
API	<u>application programming interface</u> — <ol style="list-style-type: none"> 1. A standardized set of subroutines or functions along with the parameters that a program can call. 2. A set of functions exported by a library.
asynchronous	<ol style="list-style-type: none"> 1. Hardware—A property of an event that occurs at an arbitrary time, without synchronization to a reference clock. 2. Software—An action or event that occurs at an unpredictable time with respect to the execution of a program.
attenuation	The reduction of a voltage or acoustical pressure. Measured referenced to the original voltage.
attenuator	An electronic device that reduces (attenuates) a voltage by a given amount. Attenuation is normally given in dB.
<u>averaging</u>	Averaging is a technique to improve frequency domain analysis accuracy. The principle is to average continuous frequency spectrums

	calculated out of continuous time domain waveform records. Averaging reduces noise and enhances frequency domain features for a tested system over time. Different averaging modes and averaging weighting are normally used in signal processing. For more information on the basic concepts of averaging, refer to Improving Accuracy through Averaging at zone.ni.com .
<u>B</u>	
backplane	An assembly, typically a printed circuit board (PCB), with 96-pin connectors and signal paths that bus the connector pins.
bandwidth	The range of frequencies present in a signal, or the range of frequencies to which a measuring device can respond.
bit	One binary digit, either 0 or 1.
block diagram	A pictorial description or representation of a program or algorithm. In LabVIEW, the block diagram consists of executable icons called nodes and wires that carry data between the nodes; the block diagram is the source code for the VI.
buffer	Temporary storage for acquired or generated data (software).
bus	The group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC buses are the PCI bus, AT bus, and EISA bus.
byte	Eight related bits of data, an 8-bit binary number. Also used to denote the amount of memory required to store one byte of data.
<u>C</u>	
calibration	A means of verifying and adjusting the accuracy of a device.

CMRR	<p>common-mode rejection ratio—a measure of the capability of an instrument to reject a signal that is common to both input leads. For instance, if you measure a thermocouple in a noisy environment, the noise from the environment appears on both input leads. Therefore, this noise is a common-mode voltage signal that is rejected by an amount equal to the CMRR of the instrument. The CMRR is defined by the following equation:</p> $\text{CMRR} = 20 \log(\text{Differential Gain/ Common Mode Gain})$ <p>The ratio is important because it indicates how much of the common mode signal appears in your measurement. The value of the CMRR depends on signal frequency as well and must be specified as a function of frequency. An equivalent equation to represent CMRR is as follows:</p> $20 \log(\text{Measured Common Voltage/ Applied Common Voltage})$
coercion	The automatic conversion LabVIEW performs to change the numeric representation of a data element.
<u>D</u>	
data transfer	A technique for moving digital data from one system to another.
DC	Direct current
DC-coupled	Allowing the transmission of both AC and DC signals.
DDC mode	An operating mode in which certain digitizers can be used as a digital downconverter, allowing you to zoom in on data. This mode reduces the amount of data transferred into memory.
<u>DDC</u>	digital downconverter—a digital signal processing (DSP) chip that mixes, filters, and

	decimates sampled data. In essence, the DDC allows you to zoom in on a band of frequencies much narrower than the Nyquist band of the ADC. The lower sample rate means that signals of longer duration can be stored in the same amount of memory. For spectral analysis, a smaller, faster FFT can be used to look at only the band passed through the DDC.
<u>dead time</u>	A hardware specification relevant to multiple record acquisitions. Dead time is the total time between records that the digitizer is not acquiring. This time is required for the digitizer hardware to set up for the next record. For SMC-based digitizers, it is the time from the last post-reference trigger sample of the previous record to the first minimum pre-reference trigger sample leading into the next record.
digital trigger	A TTL level signal having two discrete levels—a high and a low level.
DLL	dynamic link library
driver	Software that controls a specific hardware device, such as DAQ devices, GPIB interface devices, PLCs, RTUs, and other I/O devices.
dynamic range	The dynamic range of a signal is the ratio of the strongest, or loudest part to the weakest, or softest, part of the signal, expressed in dB.
<u>E</u>	
EEPROM	electronically erasable programmable read-only memory—ROM that can be erased with an electrical signal and reprogrammed.
equivalent-time sampling (ETS)	The process of repeatedly sampling a repeating analog signal at different times in the waveform to get a high-time resolution (effective sampling rate). See also RIS .
extrema	The minimum and maximum values.
<u>F</u>	

filtering	A type of signal conditioning that allows you to attenuate unwanted portions of the signal you are trying to measure.
<u>foldover frequency</u>	One half of the sample rate.
frequency	The basic unit of rate, measured in events or oscillations per second using a frequency counter or spectrum analyzer. Frequency is the reciprocal of the period of a signal.
<u>fundamental frequency</u>	The intended frequency of operation.
<u>G</u>	
gain	The factor by which a signal is amplified, often expressed in dB. Gain as a function of frequency is commonly referred to as the magnitude of the frequency response function.
gain accuracy	A measure of deviation of the gain of an amplifier from the ideal gain. Also called gain error.
GPIB	General Purpose Interface Bus—Synonymous with HP-IB. With the spin-off of the instrument division of Hewlett Packard, the term HP-IB is no longer used. The standard bus used for controlling electronic instruments with a computer. Also called IEEE 488 bus because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1987.
<u>H</u>	
<u>harmonic</u>	In a complex signal, a component whose frequency is a multiple of the fundamental frequency.
<u>hysteresis</u>	Lag between making a change and the effect of the change.
<u>I</u>	

interrupt	A computer signal indicating that the CPU should suspend its current task to service a designated activity.
interrupt latency	The delay between the time hardware asserts an interrupt and when the interrupt service routine is activated.
ISA	Denotes a common expansion bus used in PCs: the 8-bit and 16-bit bus design featured in the IBM PC/AT.
IVI	Interchangeable Virtual Instrument—A software standard for creating a common interface (API) to common test and measurement instruments.
<u>J</u>	
jitter	The rapid variation of a clock or sampling frequency from an ideal constant frequency.
<u>L</u>	
LabVIEW	Laboratory Virtual Instrument Engineering Workbench—A program development application based on the programming language G and used commonly for test and measurement purposes.
library	A file containing compiled object code modules, each comprised of one or more functions, that can be linked to other object code modules that make use of these functions.
loading	The effect of connecting the output of one electronic device to the input of another, such as the connection of a unit under test to an instrument. The loading effect is the difference in the output value (for example, voltage) with and without the load. To prevent loading effects, the impedance of the load must be smaller than the output impedance of the source.

<u>L-Pad</u>	Refers to the basic structure of the network: one shunt and one series branch, which gives an arrangement in the shape of an L.
<u>M</u>	
master/slave	A type of network connection in which a request is transmitted to one or more destination nodes, and those nodes send a response back to the requesting node. In industrial applications, the responding (slave) device is usually a sensor or actuator, and the requesting (master) device is usually a controller.
measurement	The quantitative determination of a physical characteristic. In practice, measurement is the conversion of a physical quantity or observation to a domain where a human being or computer can determine the value.
<u>minimum rearm time</u>	A term describing behavior pertinent to multiple record acquisitions, also known as retriggerable acquisitions. When triggering a digitizer as quickly as possible, minimum rearm time is the time from one detected reference trigger to the next when the record length is at minimum (for example, 1 sample). Rearm time is a deterministic hardware specification, but varies based on digitizer configuration.
<u>N</u>	
normal mode	For the PXI/PCI-5922, an operating mode in which the digitizer is used like a conventional oscilloscope.
Nyquist frequency	When an analog signal is sampled at a rate more than twice that of its highest frequency component, it can be properly reconstructed when reconverted back to the analog domain. The required sampling rate is called the Nyquist frequency.

O

offset	The unwanted DC voltage due to amplifier offset voltages added to a signal.
offset error	A constant error added to a measurement along the whole transfer curve. For example, in $\mathbf{mx+b}$, the offset error is b .
ohm	The measure of resistance. One ohm is the resistance through which one ampere current flows at 1 V.
OSP	onboard signal processing
oscilloscope	An instrument that displays the time domain waveform(s) of one or more signals.
oversampling factor	Multiples of the maximum real-time sampling rate.
overshoot	The amount that a square wave or transient signal exceeds the desired level when transitioning from one voltage to another.

P

passband	The range of frequencies that a device can properly propagate or measure.
PCI	peripheral component interconnect—A high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA; it is a standard for PCs and workstations and offers a theoretical maximum transfer rate of 132 Mbytes/s.
peak-to-peak	A measure of signal amplitude. The difference between the highest and lowest levels of the signal.
PFI	programmable function input
PLL	phase-locked loop—An electronic circuit which forces an output frequency to be locked to the same phase as a reference frequency.

posttriggering	The technique used on a device to acquire a programmed number of samples after trigger conditions are met.
pretriggering	The technique used on a DAQ device to keep a continuous buffer filled with data, so that when the trigger conditions are met, the sample includes the data leading up to the trigger condition.
PXI	PCI eXtensions for Instrumentation—A modular, computer-based instrumentation platform.
Q	
<u>quantization noise</u>	Noise introduced when a signal is digitized. Digitized signals always contain quantization noise because the resolution of the ADC is finite.
R	
range	The minimum and maximum analog signal levels that the ADC can digitize.
record	A collection of samples acquired relative to a trigger. For example, if a single trigger occurs and you acquire 1,000 samples, you have a single record of 1,000 samples. You can also perform multirecord acquisitions with multiple triggers. Each trigger initiates the acquisition of a record, provided that the trigger is correctly recognized by the device.
<u>resampler/halfband filter</u>	A polyphase filter that allows the output sample rate to have a non-integer relationship to the input sample rate. In essence, it acts as a fixed interpolation filter followed by an NCO controlled decimator.
resolution	The smallest signal increment that can be detected by a measurement system; resolution can be expressed in bits, in proportions, or in percent of full scale. For example, a system has

	12-bit resolution, one part in 4,096 resolution, and 0.0244 percent of full scale.
<u>RIS</u>	random interleaved sampling—A method of increasing effective sample rate by repetitively sampling a waveform.
<u>rms</u>	root mean square—A measure of signal amplitude. rms refers to the most common mathematical method of defining the effective voltage or current of an AC signal.
RTSI	real-time system integration bus—The NI timing bus that interconnects data acquisition devices directly by means of connectors on top of the devices for precise synchronization of functions.
<u>S</u>	
S	sample
s	seconds
sample	Single analog or digital input or output data point.
sample rate	Time interval between observations in a periodic sampling control system.
SNR	signal-to-noise ratio; the ratio of total signal to noise expressed in decibels (dB). The larger the number, the better. SNR is calculated by $SNR = 20 \log (\text{SignalRMS} / \text{NoiseRMS})$. This can be a peak, rms, or other amplitude that appropriately characterizes the data.
S/s	samples per second. Expresses the rate at which a device measures a signal.
synchronous	<ol style="list-style-type: none"> 1. Hardware—A property of an event that is synchronized to a reference clock. 2. Software—A property of a function that begins an operation and returns only when the operation is complete. A synchronous process is, therefore,

	"locked" and no other processes can run during this time.
<u>I</u>	
TDC	time-to-digital conversion
throughput rate	The data, measured in bytes/s, for a given continuous operation, calculated to include software overhead.
time stamps	The use of a free running clock to accurately record the time of an event.
TIS	time interleaved sampling
transfer rate	The rate, measured in bytes/s, at which data is moved from source to destination after software initialization and set up operations; the maximum rate at which the hardware can operate.
trigger	<ol style="list-style-type: none"> 1. Any event that causes, starts, or stops some form of data capture. 2. An external stimulus that initiates one or more instrument functions. Trigger stimuli include a front panel button, an external input voltage pulse, or a bus trigger command. The trigger may also be derived from attributes of the actual signal to be acquired, such as the level and slope of the signal.
trigger hysteresis	Refer to hysteresis .
<u>V</u>	
V	volts
<u>W</u>	
wave	A pattern that repeats over time.

waveform

A graphical display of a wave. A voltage waveform shows time on the horizontal (**x**) axis and voltage on the vertical (**y**) axis.

NI Services

National Instruments provides global services and support as part of our commitment to your success. Take advantage of product services in addition to training and certification programs that meet your needs during each phase of the application life cycle; from planning and development through deployment and ongoing maintenance.

To get started, create an account or log in to your National Instruments User Profile at ni.com/myproducts and register your product.

As a registered NI product user, you are entitled to the following benefits:

- Access to applicable product services.
- Easier product management with an online account.
- Receive critical part notifications, software updates, and service expirations.

Services and Resources

- **Maintenance and Hardware Services**—NI helps you identify your systems' accuracy and reliability requirements and provides warranty, sparing, and calibration services to help you maintain accuracy and minimize downtime over the life of your system. Visit ni.com/services for more information.
 - **Warranty and Repair**—All NI hardware features a one-year standard warranty that is extendable up to five years. NI offers repair services performed in a timely manner by highly trained factory technicians using only original parts at a National Instruments service center.
 - **Calibration**—Through regular calibration, you can quantify and improve the measurement performance of an instrument. NI provides state-of-the-art calibration services. If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.

- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- **Training and Certification**—The NI training and certification program is the most effective way to increase application development proficiency and productivity. Visit ni.com/training for more information.
 - The Skills Guide assists you in identifying the proficiency requirements of your current application and gives you options for obtaining those skills consistent with your time and budget constraints and personal learning preferences. Visit ni.com/skills-guide to see these custom paths.
 - NI offers courses in several languages and formats including instructor-led classes at facilities worldwide, courses on-site at your facility, and online courses to serve your individual needs.
- **Technical Support**—Support at ni.com/support includes the following resources:
 - **Self-Help Technical Resources**—Visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
 - **Software Support Service Membership**—The Standard Service Program (SSP) is a renewable one-year subscription included with almost every NI software product, including NI Developer Suite. This program entitles members to direct access to NI Applications Engineers through phone and email for one-to-one technical support, as well as exclusive access to online training modules at ni.com/self-paced-training. NI also offers flexible extended contract options that guarantee your SSP benefits are available without interruption for as long as you need them. Visit ni.com/ssp for more information.

- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification.

For information about other technical support options in your area, visit ni.com/services or contact your local office at ni.com/contact.

You can also visit the **Worldwide Offices** section of ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.