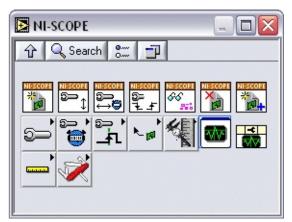
NI-SCOPE

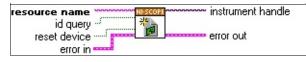
Use the VIs on the NI-SCOPE palette to build the block diagram. Click the icons for VI and function descriptions.



niScope Initialize

Performs the following initialization actions:

- Creates a new IVI instrument driver session.
- Opens a session to the device that you specify in the **resource name** parameter.
- Queries the instrument ID and checks that it is valid for this instrument driver, if the **id query** parameter is set to True.
- Resets the instrument to a known state if the **reset device** parameter is set to True; refer to <u>niScope Reset</u> for the default state of each digitizer.
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.
- Returns an instrument handle that you use to identify the instrument in all subsequent instrument driver VI calls.



resource name specifies the device name assigned by Measurement & Automation Explorer (MAX).

Examples

Example	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1 (1 = device number)
2	NI-DAQmx device	myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
4	NI-DAQmx device	DAQ::2 (2 = device name)
5	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)

For Traditional NI-DAQ devices, the syntax is DAQ::*n*, where *n* is

the device number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 2. 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 3. 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 4.

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 5. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

- Caution Traditional NI-DAQ and 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.
- **id query** verifies that the digitizer you initialize is one that NI-SCOPE supports. NI-SCOPE automatically performs this query, so setting this parameter is not necessary.
- **reset device** specifies whether to reset the instrument during the initialization procedure.
 - Note For the NI 5112, repeatedly resetting the device may cause excessive wear on the electromechanical relays.

Refer to <u>NI 5112 Electromechanical Relays</u> for recommended programming practices.

Default Value: TRUE

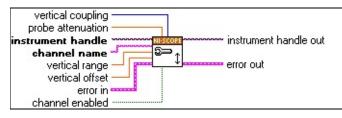
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle** identifies a particular instrument session.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>Features Supported by Device</u> for more information.



- vertical coupling specifies how to couple the input signal. When input coupling changes, the input stage takes a finite amount of time to settle.
- **probe attenuation** specifies the probe attenuation for the input channel.

Default Value: 1

- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- 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.

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of **error in** in **error out**. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the **error in** value to **error out**. The **error in** cluster contains the following parameters:

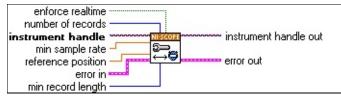
- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Configure Horizontal Timing

Configures the common properties of the horizontal subsystem for a multirecord acquisition in terms of minimum sample rate.



enforce realtime indicates whether the digitizer enforces real-time measurements or allows equivalent-time (RIS) measurements.

Not all digitizers support RIS—refer to <u>Features Supported by</u> <u>Device</u> for more information.

Default Value:TRUE

Defined Values

TRUE (Default)—Allow real-time acquisitions only

FALSE—Allow real-time and RIS acquisitions

- **number of records** specifies the number of records to acquire. Default Value: 1
- **instrument handle** identifies a particular instrument session.

min sample rate specifies the sample rate for the acquisition. Default Value: 20 MS/s

reference position specifies the position of the Reference Event in the waveform record as a percentage of the record.

Default Value: 50%

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- min record length passes the minimum number of points you need in the record for each channel.

Call <u>niScope Actual Record Length</u> to obtain the actual record length used. Refer to <u>Coercions of Horizontal Parameters</u> 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

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Configure Trigger (poly)

Configures the digitizer for different types of triggering.

When you initiate an acquisition, the digitizer waits for the start trigger, which is configured through the <u>Acquisition Arm Source (Start Trigger</u> <u>Source</u>) property. 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 Configure Trigger VI. Upon receiving the reference trigger, the digitizer finishes the acquisition after completing posttrigger sampling. With each Configure Trigger VI, 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 pretrigger and posttrigger samples using the **reference position** parameter in <u>niScope Configure Horizontal Timing</u>. The default is half of the record length.
- Some features are not supported by all digitizers. Refer to <u>Features Supported by Device</u> for more information.

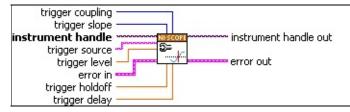
Use the pull-down menu to select an instance of this VI.

Select an instance

•

Analog Edge Ref Trigger

Configures common properties for edge triggering.



trigger coupling specifies how to couple the trigger signal.

Refer to Trigger Coupling for defined values.

trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to Trigger Slope 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 Trigger Source for defined values.

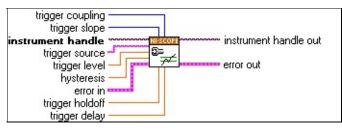
- **trigger level** specifies the voltage threshold for the trigger. Default Value: 0.0 V
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.



twinners in a laboff and a life a the allowards of times the adjustice systematic

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.



- **trigger coupling** specifies how to couple the trigger signal. Refer to <u>Trigger Coupling</u> for defined values.
- **trigger slope** specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to Trigger Slope 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 Trigger Source for defined values.

hysteresis specifies the size of the hysteresis window on either side of the trigger 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 the trigger slope, and passes through the trigger level.

Default Value: 0.05 V

Valid Values:

Min Value: 0

Max Value for positive trigger slope:

```
Hysteresis – trigger level >= –(vertical range/2) + vertical offset
Max value for negative trigger slope:
```

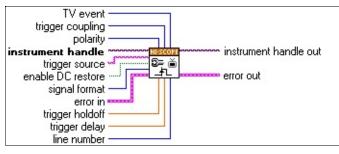
Hysteresis + *trigger level* <= (*vertical range*/2) + *vertical offset*



trigger level specifies the voltage threshold for the trigger.

Video Ref Trigger

Configures the common properties specific to video triggering, such as trigger coupling, trigger source, trigger holdoff, and trigger delay. 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 1 frame in NTSC (121 ms < holdoff < 159 ms) and 5 frames in SECAM (201 ms < holdoff < 239 ms).



TV event specifies what TV event to trigger on.

Refer to the **Event** property for defined values.

trigger coupling specifies how to couple the trigger signal. Refer to <u>Trigger Coupling</u> for defined values.

polarity specifies the polarity of the video sync.

Default Value: negative

Defined Values:

positive

negative

- **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 Trigger Source for defined values.

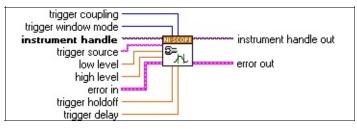
enable DC restore offsets each video line so the clamping level (the portion of the video line between the color burst and the beginning of the active image) is moved to zero volt.

Default Value: FALSE



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.



trigger coupling specifies how to couple the trigger signal. Refer to <u>Trigger Coupling</u> for defined values.

trigger window mode specifies whether the trigger should occur when the signal is entering or leaving a window.

Default Value: Entering

- **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 Trigger Source for defined values.

Iow level passes the voltage threshold for the digitizer to use for low triggering.

Default Value: 0 V

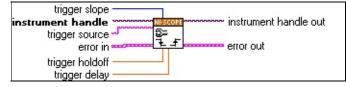
high level passes the voltage threshold for the digitizer to use for high triggering.

Default Value: 0.10 V

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

Digital Edge Ref Trigger

Configures the common properties of a digital trigger.



trigger slope specifies either a rising edge or a falling edge to trigger the digitizer.

Refer to Trigger Slope 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 Trigger Source for defined values.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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.

Immediate Ref Trigger

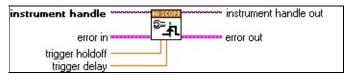
Configures common properties for immediate triggering. Immediate triggering means the digitizer triggers itself.

instrument handle

- **instrument handle** identifies a particular instrument session.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- instrument handle out has the same value as the instrument handle.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Software Ref Trigger

Configures common properties for software triggering. This VI only works in an Initiate/Fetch operation.



instrument handle identifies a particular instrument session.

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.

status is TRUE (X) if an error occurred or FALSE

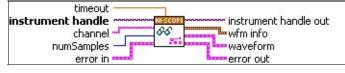
niScope Read (poly)

Initiates an acquisition, waits for it to complete, and acquires data. Use the pull-down menu to select an instance of this VI.

Select an instance 🚽

Cluster

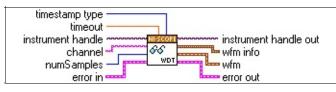
Initiates an acquisition, waits for it to complete, and retrieves the data for a single channel and record.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

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.



timestamp type specifies the time basis for the timestamp on the WDT data.

Defined Values

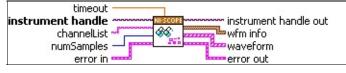
absolute

relative

- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If

1D Cluster

Initiates an acquisition, waits for it to complete, and retrieves the data for multiple channels and records.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

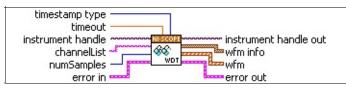
- **numSamples** is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use –1 for this parameter if you want to fetch all available 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.
- **error in** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of **error in** in **error out**. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the **error in** value to **error out**. The **error in** cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.



instrument handle out has the same value as the instrument

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.



timestamp type specifies the time basis for the timestamp on the WDT data.

Defined Values

absolute

relative

- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

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.



instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a

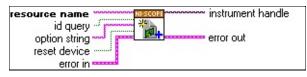
warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Initialize with Options

Performs the following initialization actions:

- Creates a new IVI instrument driver and optionally sets the initial state of the following session properties: Range Check, Cache, Simulate, Record Value Coercions.
- Opens a session to the device that you specify in the **resource name** parameter.
- Resets the instrument to a known state if the **reset device** parameter is set to True; refer to <u>niScope Reset</u> for the default state of each digitizer.
- Queries the instrument ID and verifies that it is valid for this instrument driver if the **id query** parameter is set to True.
- Returns an instrument handle that you use to identify the instrument in all subsequent instrument driver VI calls.



resource name specifies the device name assigned by Measurement & Automation Explorer (MAX).

Examples

Example	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1 (1 = device number)
2	NI-DAQmx device	myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)
4	NI-DAQmx device	DAQ::2 (2 = device name)
5	IVI logical name or IVI virtual name	myLogicalName (myLogicalName = name)

For Traditional NI-DAQ devices, the syntax is DAQ::*n*, where *n* is

the device number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 2. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on he 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 3. 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 4.

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 5. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

- Caution Traditional NI-DAQ and 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.
- **id query** verifies that the digitizer you initialize is one that NI-SCOPE supports. NI-SCOPE automatically performs this query, so setting this parameter is not necessary.
- option string specifies initialization commands.

The following table lists the properties and the name you use in this parameter to identify the property:

Name	Attribute Defined Constant	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

Defined Values

TRUE (1) FALSE (0)

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 an NI PXI-5102 would be as follows:

Option String: Simulate = 1, DriverSetup = Model:5102; BoardType:PXI

Refer to the example niScope EX Simulated Acquisition for more information on simulation.



Caution All IVI names, such as logical names or virtual 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 in the name.



reset device specifies whether to reset the instrument during the initialization procedure.

Note For the NI 5112, repeatedly resetting the device may cause excessive wear on the electromechanical relays. Refer to NI 5112 Electromechanical Relays for recommended programming practices.

Default Value: TRUE

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle** identifies a particular instrument session.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

NI-SCOPE Express

Acquires an analog voltage waveform from a National Instruments highspeed digitizer.

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	Contains the following options:	
	 Hardware—Configures the overall device. 	
	 Device—Specifies the device used. Unavailable devices are disabled. 	
	 Channels—Select or multi-select channel(s) by highlighting. The channel configurations for selected channels are diplayed. Place a checkmark next to the channel(s) you want to enable. Only enabled channels will acquire data. 	
	 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. The following parameters are displayed:	
	• Range (V) —Sets the full vertical range.	
	 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 attenuation. 	
	 Coupling—Sets the coupling on the analog signal. 	
	 Input impedance—Sets the input impedance. 	
	Bandwidth (Hz)—Sets the frequency of the	

	 input low-pass 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)—Sets 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. 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	 Configures options used less frequently. Contains the following: 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 ControlContains the following options:• Pre-execution delay (ms)—Specifies the of time to wait before the step executes. It configure the step to start after another st delay represents the amount of time to wait 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 generating 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
•	Returns the data collected by the digitizer. The number of channels depends upon how many channels are enabled.
	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 Property Node

The niScope Property Node is used to set, get, or check properties.

Configuration

Use the VIs on the Configuration palette to set up the parameters of an acquisition.

Click the icons for VI and function descriptions.

🖪 Configuration 🛛 🖃 🗖 🔀				
🗘 🔍 Search	0	J		
		<mark>≆COPE</mark> ≽Ω ≂¥-348		

niScope Auto Setup

Automatically configures the digitizer. Details

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 $1.2 V_{pk-pk}$ 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.



instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Details

The following settings are changed by this VI:

General

Normal
Internal
AC (DC for NI 5621)
Full
Changed by auto setup
0 V
Unchanged by auto setup
Unchanged by auto setup
Changed by auto setup
Changed by auto setup
True
Changed to 1

Trigger mode	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

niScope Configure Acquisition

Configures how the digitizer acquires data and fills the waveform record.

instrument handle water acquisition Type

instrument handle identifies a particular instrument session.

acquisition type specifies the manner in which the digitizer acquires data and fills the waveform record.

Not all digitizers support all modes; refer to <u>Features Supported by</u> <u>Device</u> for more information.

Default Value: Normal

Defined Values

Normal

Flex Res

DDC



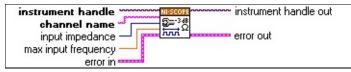
- Note Acquisition type DDC applies to the NI 5620/5621 only. To use DDC mode in the NI 5142, leave **acquisition type** set to Normal and set the <u>DDC Enabled</u> property to True.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Configure Chan Characteristics

Configures the properties that control the electrical characteristics of the channel. These properties are the input impedance and the bandwidth.



- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- **input impedance** is the input impedance for the channel.

Default Value:1 mega ohm

Defined Values

1 mega ohm

50 ohms

max 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

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

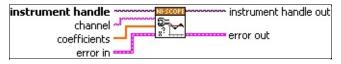
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- error out contains error information. If error 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.

 $\overline{\mathbb{N}}$

Note This VI can be used only with high-speed digitizers that support onboard signal processing (OSP).



instrument handle identifies a particular instrument session.

- **channel name** is the channel on which to configure the filter. For more information, refer to <u>Channel String Syntax</u>.
- **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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Timing

Use the VIs on the Timing palette to configure common timing properties. Click the icons for VI and function descriptions.

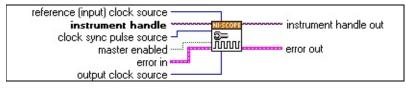


niScope Configure Clock

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 <u>Features Supported by Device</u> for more information.



input clock source specifies the input source for the PLL reference clock (the 1–20 MHz clock on SMC-based devices, the 10 MHz clock for the NI 5112/5620/5621/5911) to which the digitizer is phase-locked for all digitizers except the NI 5102; for the NI 5102, this is the source of the board clock.

Refer to the Input Clock Source property for defined values.

Default Value: None

- **instrument handle** identifies a particular instrument session.
- **Clock sync pulse source** specifies the line on which the sample clock is sent or received for the NI 5102; for the NI 5112/5620/5621/5911, specifies the line on which the one-time sync pulse is sent or received. This line should be the same for all devices to be synchronized.

Refer to <u>Clock Sync Pulse Source</u> for defined values.

Default Value:RTSI 0

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 stand-alone device, set this parameter to FALSE.

Refer to the Master Enable property for more information.

Default Value: FALSE

error in describes error conditions that occur before this VI runs.

The default input of this cluster is no error. If an error already occurred, this VI returns the value of **error in** in **error out**. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the **error in** value to **error out**. The **error in** cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **output clock source** specifies the output source for the PLL reference clock (20 MHz in the NI 5102) to which another digitizer's sample clock can be phased-locked.

For the NI 5102, set this parameter to RTSI Clock to specify the NI 5102 as a master; refer to <u>Output Clock Source</u> for more information.

Default Value: None

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

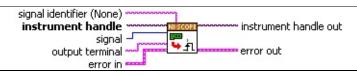
niScope Export Signal

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.





- **signal identifier** describes the signal being exported.
- **instrument handle** identifies a particular instrument session.
- **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 (9)—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 Clock Out

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Adjust Sample Clock Relative Delay

Configures the relative sample clock delay (in seconds) when using the internal clock. Each time this VI is called, the sample clock is delayed by the specified amount of time.

instrument handle	NI-SCOPE CONTRACTOR	instrument handle out
delay		
error in		error out

- **instrument handle** identifies a particular instrument session.
- 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

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Actual Values

Use the VIs on the Actual Values palette to configure actual sample mode, record length, and sample rate.

Click the icons for VI and function descriptions.

🖪 Actual Values 📃 🗆 🔀		
🗘 🔍 Search	: 1	
	2002 Տ/Տ, Եվլի	

niScope Actual Sample Mode

Returns the sample mode the digitizer is currently using.

error in

instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **sample mode** returns the sample mode the digitizer is currently using.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.



instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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 <u>Horizontal Actual Record</u> <u>Length</u> property; refer to <u>Coercions of Horizontal Parameters</u> for more information.

- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Sample Rate

Returns the effective sample rate, in samples per second, of the acquired waveform using the current configuration.

Refer to <u>Coercions of Horizontal Parameters</u> for more information about sample rate coercion.



- **instrument handle** identifies a particular instrument session.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Triggering

Use the VIs on the Triggering palette to configure common triggering properties.

Click the icons for VI and function descriptions.



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.



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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument**

handle.

[201]

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.

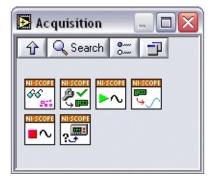
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Acquisition

Use the VIs on the Acquisition palette to control data acquisition and retrieve data from the digitizer.

Click the icons for VI and function descriptions.



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. This VI is not supported by Traditional NI-DAQ (Legacy) devices.

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 <u>niScope Initiate Acquisition</u> call.

instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

TF

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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>niScope Configure Vertical</u>.

instrument handle	www.www.weither.com
error in	error out

- **instrument handle** identifies a particular instrument session.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Fetch (poly)

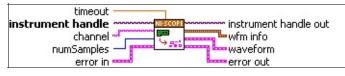
Retrieves data that the digitizer has acquired from a previously initiated acquisition.

Use the pull-down menu to select an instance of this VI.

Select an instance 🔽

Cluster

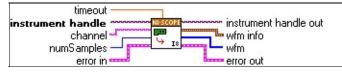
Retrieves the waveform the digitizer has acquired for the specified channel from a previously initiated acquisition. Returns scaled voltage waveforms in a cluster that includes timing information.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument**

1D I8

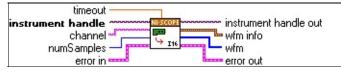
Retrieves data from a single channel and record. Returns a onedimensional array of binary 8-bit values.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

1D |16

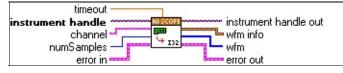
Retrieves data from a single channel and record. Returns a onedimensional array of binary 16-bit values.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

1D I32

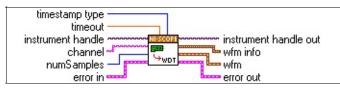
Retrieves data from a single channel and record. Returns a onedimensional array of binary 32-bit values.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

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.



timestamp type specifies the time basis for the timestamp on the WDT data.

Defined Values

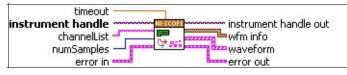
absolute

relative

- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If

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.



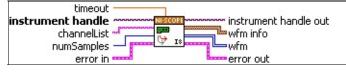
- **timeout** is the time (in seconds) to wait for the data to be acquired; use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

2D 18

Retrieves data from multiple channels and records. Returns a twodimensional array of binary 8-bit waveforms.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

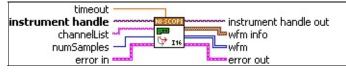
- **numSamples** is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use –1 for this parameter if you want to fetch all available 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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.



instrument handle out has the same value as the instrument

2D |16

Retrieves data from multiple channels and records. Returns a twodimensional array of binary 16-bit waveforms.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

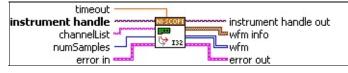
- **numSamples** is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use –1 for this parameter if you want to fetch all available 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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.



instrument handle out has the same value as the instrument

2D I32

Retrieves data from multiple channels and records. Returns a twodimensional array of binary 32-bit waveforms.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

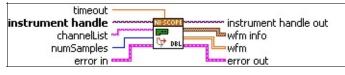
- **numSamples** is the maximum number of samples to fetch for each waveform; if the acquisition finishes with fewer points than requested, some devices return partial data if the acquisition finished, was aborted, or a timeout of 0 was used. Use –1 for this parameter if you want to fetch all available 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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.



instrument handle out has the same value as the instrument

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.



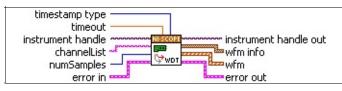
- **timeout** is the time (in seconds) to wait for the data to be acquired; use 0 for this parameter to tell NI-SCOPE to fetch whatever is currently available.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

1D 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.



timestamp type specifies the time basis for the timestamp on the WDT data.

Defined Values

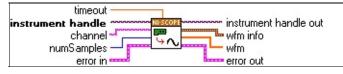
absolute

relative

- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If

1D DBL

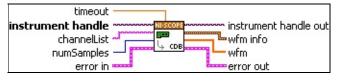
Retrieves data from a single channel and record. Returns a onedimensional array of scaled voltages.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

1D Complex DBL

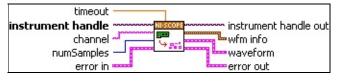
Retrieves data from single channels and records. Returns a onedimensional array of complex, scaled waveforms.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

Cluster Complex DBL

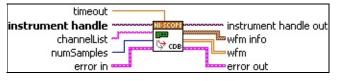
Retrieves data from single channels and records. Returns a complex, scaled waveform in a cluster that contains timing information.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

2D Complex DBL

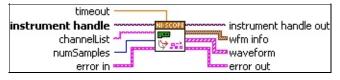
Retrieves data from multiple channels and records. Returns a twodimensional array of complex, scaled waveforms.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

1D Cluster Complex DBL

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.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **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.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument**

niScope Abort

Aborts an acquisition without changing the settings on the digitizer. Use this VI if the digitizer times out waiting for a trigger.

instrument handle

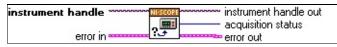
instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Acquisition Status

Returns status information indicating whether an acquisition is in progress, complete, or unknown to the **acquisition status** output parameter.



- **instrument handle** identifies a particular instrument session.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **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 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Calibration

Use the VIs on the Calibration palette to self-calibrate your digitizer or to access the external calibration VIs.

Click the icons for VI and function descriptions.



niScope Cal Self Calibrate

Self-calibrates most NI digitizers, including all SMC-based devices and most Traditional NI-DAQ (Legacy) 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 <u>niScope Cal End</u> is called with action set to Store Calibration and no errors occur.

Note To verify that your digitizer supports self-calibration, refer to <u>Features Supported by Device</u>.

instrument handle	MISCOPP miscopp
Channel Name	
Option	error out
error in	

- **instrument handle** identifies a particular instrument session.
- **Channel Name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

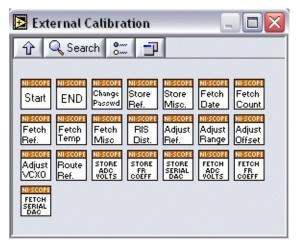
is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

External Calibration

Use the VIs on the External Calibration palette to externally calibrate the digitizers. These VIs are intended to be used with the manual calibration procedure for each device.

Click the icons for VI and function descriptions.



niScope Cal Adjust Compensation Attenuator

For the NI 5132/5133, this function externally calibrates the compensation attenuator. Refer to the *NI 5132/5133 Calibration Procedure* for more information.



Note Use this VI only when following the *NI* 5132/5133 *Calibration Procedure*

instrument handle 😁	www.w.score
channel name 🚽	Comp Atten
range (¥) —	error out
error in 🚥	4

- **instrument handle** is the instrument handle that you obtain from <u>niScope Cal Start</u>. The handle identifies a particular instrument's session.
- **channel name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **range (V)** is the vertical range to calibrate.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Adjust Internal Reference

For the NI 5911, externally calibrates the internal reference. This VI uses the external reference source instead of the internal reference source to calibrate the vertical range and offset. The internal source voltage is measured, and this value is stored and used for subsequent selfcalibrations.



Note Use this VI only when following the *NI* 5911 Calibration *Procedure* (Start» Programs»National Instruments»NI-SCOPE»Documentation»Calibration).

instrument handle out option Adjust stimulus error out error in

- **instrument handle** identifies a particular instrument session.
- **option** is not defined.
- **stimulus** is the external stimulus applied to the digitizer. Refer to the calibration procedure document for more information.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Adjust Offset

For the NI 5102, externally calibrates a specified channel and vertical range.

Call this VI before calling <u>niScope Cal Adjust Range</u> to perform a preliminary offset calibration. The first time this VI is called, a default vertical range parameter is used in the driver to calibrate the vertical offset. When niScope Cal Adjust Offset is called again after niScope Cal Adjust Range, it performs a precise offset calibration using the calibrated vertical range.

Note Use this VI only when following the *NI* 5102 Calibration *Procedure* (Start»Programs»National Instruments»NI-SCOPE»Documentation»Calibration).



- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **range (V)** is the vertical range to calibrate.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Adjust Offset Range

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.



Note Use this VI only when following the *NI* 5132/5133 *Calibration Procedure*.

instrument handle channel name range (V) stimulus	Adjorf Range error out
error in 🚥	

- **instrument handle** is the instrument handle that you obtain from <u>niScope Cal Start</u>. The handle identifies a particular instrument's session.
- **channel name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument**

handle.

[201]

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.

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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Adjust Range

Use this VI only when following the external calibration procedure for your device. (Start»Programs»National Instruments»NI-SCOPE»Documentation»Calibration). Use the VI parameters to indicate the voltage and channel.

• NI 5102

For the NI 5102, externally calibrates the vertical range for the specified channel and vertical range setting. Call <u>niScope Cal</u> <u>Adjust Offset</u> prior to calling this VI, using the same channel name and range inputs, to perform a preliminary offset calibration using a default vertical range.

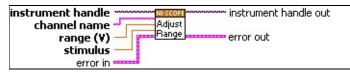
SMC-Based Devices

For SMC-based digitizers, calibrates the range (that is, gain) accuracy of the entire device using the specified channel. The VI 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 VI 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 <u>niScope Cal</u> <u>End</u> with no errors and with **action** set to Store Calibration.

• NI 5911

For the NI 5911, externally calibrates the AC gain for the specified channel.



- **instrument handle** is the instrument handle that you obtain from <u>niScope Cal Start</u>. The handle identifies a particular instrument's session.
- **channel name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **range (V)** is the vertical range to calibrate
- **stimulus** is the external stimulus applied to the digitizer
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE if an error occurred; the default is FALSE
 - **code** is the error code; the default is 0
 - **source** is, in most cases, the name of the VI or function that produced the error; the default is an empty string
- **instrument handle out** has the same value as the **instrument handle**.
- error out contains error information; if the error in cluster indicated an error, the error out cluster contains the same information; otherwise, error out describes the error status of this VI
 - **status** is set to TRUE if an error occurred
 - **code** is the error code
 - **source** is, in most cases, the name of the VI or function that produced the error

niScope Cal Adjust VCXO

For SMC-based devices, 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. The **stimulusFreq** parameter must be set to 10,000,000 (10 MHz). Before calling this VI, connect an accurate, stable 10 MHz reference to channel 0. (The channel used is not configurable.) The VI adjusts frequency calibration constants until the digitizer measures the 10 MHz reference signal as exactly 10 MHz. The new calibration constants take effect immediately for the duration of the external calibration session. The constants are written to EEPROM if you call <u>niScope Cal End</u> with no errors and with **action** set to Store Calibration.



Note Use this VI only when following the *NI* 5122/5124 *Calibration Procedure* (Start»Programs»National Instruments»NI-SCOPE»Documentation»Calibration).

instrument handle
instrument handle out
stimulus frequency (Hz)
error in

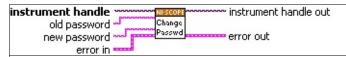
- **instrument handle** identifies a particular instrument session.
- **stimulus frequency (Hz)** is the frequency of the external reference clock connected to channel 0; set to 10,000,000 (10 MHz).
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.



- **instrument handle** identifies a particular instrument session.
- **old password** is the password currently stored in the EEPROM.
- **new password** is the new password to store in the EEPROM. A maximum of 4 characters can be stored.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal End

Closes an external calibration session. If **action** is set to Abort Calibration, 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 this VI each time you call <u>niScope Cal Start</u>, even if an error occurs during calibration.

If **action** is set to Store Calibration, 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 devices, the current system date and onboard temperature are stored.

instrument handle	 NI-SCOPE	
action	 END	Contraction of the second sec second second sec
error in	 	error out

- **instrument handle** identifies a particular instrument session.
- **action** either stores the calibration constants or aborts the calibration and discards any calibration results.

Defined Values

Store Calibration

Abort Calibration

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

what inputs are in error, and how to eliminate the error.

- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Fetch Count

Returns the number of times the digitizer has been calibrated.

instrument handle
Which count
Fetch
count
error in
Count
error out

- **instrument handle** identifies a particular instrument session.
- which count specifies which calibration count to return (selfcalibration or external calibration).

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

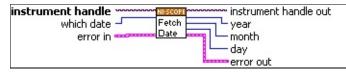
- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Fetch Date

Returns the last self-calibration, external calibration, or manufacture date.



instrument handle identifies a particular instrument session.

which date is which calibration date to return (self-calibration, external calibration, or manufacture date).

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **year** is the year of the last calibration, such as 2003.
- **month** is the month of the last calibration (1-12).
- **Data** day is the day of the last calibration (1–31).

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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Fetch Internal Reference

Returns the value of the internal reference that is stored with <u>niScope Cal</u> <u>Store Internal Reference</u> during external calibration. This VI allows tracking of the internal reference value, although the value is not used during digitizer operation or self-calibration. The internal reference value is stored as a 32-bit floating point number in the EEPROM.

instrument handle	NESCOPE	instrument handle out
micritorororico	Fetch	internal reference
error in	Ref.	error out

- **instrument handle** identifies a particular instrument session.
- which reference specifies which reference to store, fetch, or route.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **internal reference** is the value of the internal reference stored during the last external calibration.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Fetch Misc Info

Returns the miscellaneous information you can store during an external calibration using <u>niScope Cal Store Misc Info</u>.

instrument handle vouse Miscolla miscellaneous information Fetch miscellaneous information error in miscellaneous information

instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **miscellaneous information** is four characters that are stored in the EEPROM; however, it can be fewer than four characters if NULL-terminated.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Fetch Temperature

For SMC-based devices, 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.

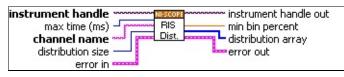
instrument handle	NI-SCOPE	instrument handle out
	Fetch	temperature (Celsius)
error in	Temp	error out

- **instrument handle** identifies a particular instrument session.
- 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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.



- **instrument handle** identifies a particular instrument session.
- **max time (ms)** is the maximum time to allow for each acquisition.
- **channel name** is the channel to calibrate. For more information, refer to <u>Channel String Syntax</u>.
- **distribution size** is the number of bins in the **initial x** distribution.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument**

handle.

- **min bin percent** is the percent of triggers in the minimum bin (0– 100.0).
- **distribution array** (waveform array) is the returned distribution of trigger times.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Route Internal Reference

For the NI 5112, routes the internal reference to the front BNC connector of the specified channel. Use this VI when measuring the internal reference of the NI 5112 and storing the value in the EEPROM with <u>niScope Cal Store Internal Reference</u>, which allows tracking of the verification procedure and tracking the drift of the source over time.

instrument handle	NI-SCOPE	instrument handle out
which reference	Route	
options	Bef	error out
error in	8	

- **instrument handle** identifies a particular instrument session.
- which reference specifies which reference to store, fetch, or route.
- **options** specifies options for routing the signal.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

TF	status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
132	code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
labc	source identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>niScope Initialize</u>. Acceptable session handles are documented for each VI in the manual calibration procedure document for each device.

By default, the calibration password for SMC-based devices is "NI"; the default password for all other digitizers is an empty string. 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 legal passwords. The password is verified against the password stored in the EEPROM. You can change the password from the default by calling niScope Cal Change Password.

Only one session handle can be obtained at a time, and every session started with this VI must be closed by calling <u>niScope Cal End</u>. 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	NI-SCO	instrument handle
password	Star	t
error in		error out

resource name specifies the device name assigned by Measurement & Automation Explorer (MAX).

Example	Device Type	Syntax
1	Traditional NI-DAQ device	DAQ::1 (1 = device number)
2	NI-DAQmx device	myDAQmxDevice (myDAQmxDevice = device name)
3	NI-DAQmx device	DAQ::myDAQmxDevice (myDAQmxDevice = device name)

Examples

4	NI-DAQmx device	DAQ::2 (2 = device name)
	•	myLogicalName (myLogicalName = name)

For Traditional NI-DAO devices, the syntax is DAO::*n*, where *n* is the device number assigned by MAX, as shown in Example 1.

For NI-DAQmx devices, the syntax is just the device name specified in MAX, as shown in Example 2. Typical default names for NI-DAQmx devices in MAX are Dev1 or PXI1Slot1. You can rename an NI-DAQmx device by right-clicking on he 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 3. 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 4.

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 5. A logical name identifies a particular virtual instrument. A virtual name identifies a specific device and specifies the initial settings for the session.

Caution Traditional NI-DAQ and 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.

abc **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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle** identifies a particular instrument session.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Cal Store Internal Reference

For the NI 5112, this VI allows you to store the measured internal reference voltage value.

The NI 5112 supports routing the internal reference to the front BNC connectors, using <u>niScope Cal Route Internal Reference</u>. You can measure the internal reference with a DMM to verify it is within specification, then call this VI to store the measured value in the EEPROM. Storing the internal reference allows you to track the drift of the internal reference over time. The reference value is stored for tracking purposes only, and is not used during digitizer operation. The reference value is stored in the EEPROM as a 32-bit floating point number.

This VI requires an instrument handle returned from <u>niScope Cal Start</u>. The reference value is not written to the EEPROM until <u>niScope Cal End</u> is called with **action** set to Store Calibration. If this VI is not called during an external calibration session, the internal reference value is set to zero in the EEPROM when niScope Cal End is called with the **action** set to Store Calibration. This ensures consistent calibration count, date, and internal reference values in the EEPROM.

instrument handle •	NESCOPE	instrument handle out
which reference -		
internal reference -	Bef.	error out
error in •		

- **instrument handle** identifies a particular instrument session.
- which reference specifies which reference to store, fetch, or route.
- **internal reference** is the value of the internal reference stored during the last external calibration.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **error in** describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>niScope Initialize</u> or <u>niScope Cal Start</u>. The handle identifies a particular instrument's 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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE if an error occurred; the default is FALSE
 - **code** is the error code; the default is 0
 - **source** is, in most cases, the name of the VI or function that produced the error; the default is an empty string
- **instrument handle out** has the same value as the **instrument handle**.
- error out contains error information; if the error in cluster indicated an error, the error out cluster contains the same information; otherwise, error out describes the error status of this VI
 - **status** is set to TRUE if an error occurred
 - **code** is the error code

source is, in most cases, the name of the VI or function that produced the error

Measurements

Use the VIs on the Measurements palette to perform waveform measurements, including scalar and array measurements.

Click the icons for VI and function descriptions.

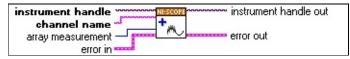


niScope Add Waveform Processing

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 attribute **filter type** to lowpass, and then register a Bessel Filter as a processing step using this function. Then you can set the **filter type** to bandpass and register a Chebyshev filter. In a loop, call <u>niScope Read Measurement</u> 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 <u>Array Measurements</u> for more information about the available measurements.



- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- array measurement is the array measurement to add as a processing step.

Refer to the list of array measurements for more information.

Default Value: None

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>Measurement Chan Based Low Ref</u>, <u>Measurement Chan</u> <u>Based Mid Ref</u>, and <u>Measurement Chan Based High Ref</u> to set each channel differently.

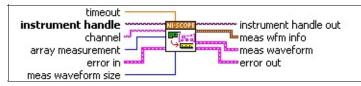
Use the pull-down menu to select an instance of this VI.

•

Select an instance

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.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- array measurement is the measurement to perform.

Refer to Array Measurements for more information.

- **error in** describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of **error in** in **error out**. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the **error in** value to **error out**. The **error in** cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.

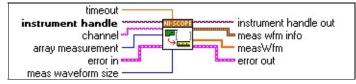
Default Value: -1 (returns all available samples)



Note Use the property <u>Fetch Meas Num Samples</u> to set

Measurement 1D DBL

Obtains a waveform from the digitizer and returns the specified measurement voltage data for a single channel and record.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- array measurement is the measurement to perform.

Refer to Array Measurements for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.

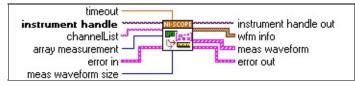
Default Value: -1 (returns all available samples)



Note Use the property <u>Fetch Meas Num Samples</u> to set the number of samples to fetch when performing a

Measurement 1D Cluster

Obtains a waveform from the digitizer and returns the specified array measurement for multiple channels and records as a cluster.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

array measurement is the measurement to perform.

Refer to <u>Array Measurements</u> for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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

Measurement 2D DBL

timeout	
instrument handle	
channelList 🚽 👝 🔤 🛄 🚾 wfm info	
array measurement — 🚰 🚟 🔤 🗠 wfm	
error in	
meas waveform size ————	

- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

array measurement is the measurement to perform.

Refer to Array Measurements for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- meas waveform size is the maximum number of samples returned in the measurement waveform array for each waveform measurement.

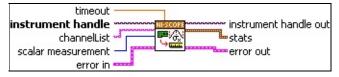
Default Value: -1 (returns all available samples)



Note Use the property <u>Fetch Meas Num Samples</u> to set the number of samples to fetch when performing a measurement. For more information about when to use

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.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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-topeak. 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.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - abc
- source identifies where an error occurred. The source

Measurement Scalar DBL

Obtains a waveform from the digitizer and returns the specified measurement voltage data for a single channel and record.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **scalar measurement** is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- 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 <u>Fetch Meas Num Samples</u> to set

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.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **results** contains the measurements acquired as a one-

niScope Clear Waveform Processing

Clears the list of processing steps assigned to the given channel.

The processing steps are added with <u>niScope Add Waveform</u> <u>Processing</u>, 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.



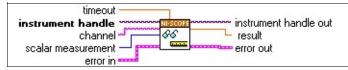
- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Read Measurement

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for a single channel and record.

Use <u>niScope Multi Read Measurement</u> for multiple records and channels.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channel** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.
- **scalar measurement** is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- **result** contains the measurement acquired.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.

instrument handle	instrument handle out
channel name	
clearable measurement	error out
error in	

- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- **clearable measurement** is the measurement for which to clear the statistics.

Refer to <u>Scalar Measurements</u> or <u>Array Measurements</u> for constants. To clear all of the measurements, specify All Measurements.

Default Value: All Measurements

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

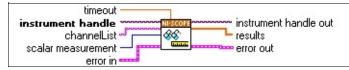
what inputs are in error, and how to eliminate the error.

- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Multi Read Measurement

Initiates an acquisition, waits for it to complete, and performs the specified waveform measurement for multiple channels and records.

Use <u>niScope Read Measurement</u> for a single channel and record.



- **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.
- **instrument handle** identifies a particular instrument session.
- **channelList** is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

scalar measurement is the measurement to perform on the waveform read from the digitizer.

Refer to the list of <u>scalar measurements</u> available for more information.

- **error in** describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.

results contains the measurements acquired as a onedimensional 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.

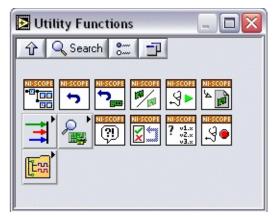
For example, with a two-channel list, you would have the following index values:

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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Utility Functions

Use the Utility Functions palette to perform various tasks such as resetting the digitizer and returning the revision numbers of the driver and the instrument firmware.



niScope Reset

Resets the digitizer to its default state. Refer to <u>Property Defaults</u> for the default property values of each device.

instrument handle identifies a particular instrument session.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.



instrument handle identifies a particular instrument session.

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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	Instrument handle out	
error in	error out	

- **instrument handle** identifies a particular instrument session.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Probe Compensation Signal Start

Starts the 1 kHz square wave output on PFI 1 for probe compensation.

Note Some features are not supported by all digitizers. Refer to <u>Features Supported by Device</u> for more information.

instrument handle	NI-SCOPE	instrument handle out
error in		error out

instrument handle identifies a particular instrument session.

error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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.

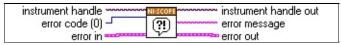


- **instrument handle** identifies a particular instrument session.
- **error in** describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 <u>niScope Initialize</u> has failed.



- **instrument handle** identifies a particular instrument session.
- error code is the status code returned by any NI-SCOPE VI.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- error message returns the interpreted error code as a userreadable string.
- 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.

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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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. 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.

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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Probe Compensation Signal Stop

Stops the 1 kHz square wave output on PFI 1 for probe compensation.

Note Some features are not supported by all digitizers. Refer to <u>Features Supported by Device</u> for more information.

instrument handle	NI-SCOPE	instrument handle out
error in		error out

instrument handle identifies a particular instrument session.

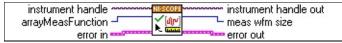
error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Actual Meas Wfm Size

Returns the total available size of an array measurement acquisition in samples.



instrument handle identifies a particular instrument session.

arrayMeasFunction is the measurement to perform.

Refer to the list of available <u>array measurements</u> for more information.

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- meas wfm size returns the size of the resulting analysis waveform in samples.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Actual Num Wfms

Allows you to declare appropriately-sized waveforms. NI-SCOPE handles the channel list parsing for you.

instrument handle and a compared to the second seco

instrument handle identifies a particular instrument session.

channelList is the channel from which to acquire data. For more information, refer to <u>Channel String Syntax</u>.

Default Value: 0

- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- **actual num wfms** returns the number of records times the number of channels; if you are operating in DDC mode (NI 5620/5621 only), this value is multiplied by two.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

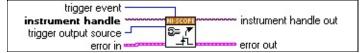
niScope Configure Trigger Output

Configures the digitizer to generate a signal pulse that other digitizers can detect when configured for digital triggering.

M **Notes** For Traditional NI-DAQ devices, exported signals are still present in the route after the session is closed. You must clear the route before closing the session, or call niScope reset. To clear the route, call this function again and route None to the line that you had exported. For example, if you originally called this function with the trigger event Stop Trigger routed to the trigger output RTSI 0, you would call this function again with None routed to RTSI 0 to clear the route.

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 <u>niScope Export Signal</u> instead.



trigger event specifies the condition in which this device will generate a digital pulse.

Refer to <u>Trigger Output Event</u> for more information.

- **instrument handle** identifies a particular instrument session.
- **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

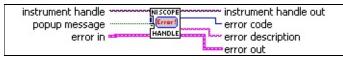
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

error, what inputs are in error, and how to eliminate the error.

niScope Error Handler

This VI is either obsolete or included for IVI compliance only. Consider using <u>niScope Error Message</u> instead. This VI takes the error code returned by NI-SCOPE VIs and returns the interpretation as a user-readable string.

You can pass VI_NULL as the instrument handle, which is useful for interpreting errors after <u>niScope Initialize</u> has failed.



instrument handle is the handle that you obtain from <u>niScope</u> <u>Initialize</u> that identifies a particular instrument's session

popup message enables a pop up message when the case is set to Yes

Default Value

No

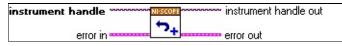
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE if an error occurred; the default is FALSE
 - **code** is the error code; the default is 0
 - **source** is, in most cases, the name of the VI or function that produced the error; the default is an empty string
- instrument handle out has the same value as the instrument handle
- error code returns the error code
- error description returns the interpreted error code as a user readable message string
- error out contains error information; if the error in cluster indicated an error, the error out cluster contains the same

information; otherwise, $\ensuremath{\text{error}}$ out describes the error status of this $\ensuremath{\mathsf{VI}}$

- **status** is set to TRUE if an error occurred
- **code** is the error code
- **source** is, in most cases, the name of the VI or function that produced the error

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 VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:

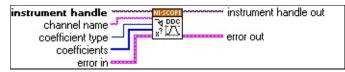
- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.
 - **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niScope Set DDC Filter Coefficients

Downloads programmable FIR and discriminator FIR filter coefficients to the digital downconverter (<u>DDC</u>) in the NI 5620/5621.



- **instrument handle** identifies a particular instrument session.
- **channel name** is the channel to configure. For more information, refer to <u>Channel String Syntax</u>.
- **coefficient type** specifies which set of coefficients to program.
- **coefficients** is the array of coefficients in 32-bit twos complement format.
- error in describes error conditions that occur before this VI runs. The default input of this cluster is no error. If an error already occurred, this VI returns the value of error in in error out. The VI runs normally only if no incoming error exists. Otherwise, the VI passes the error in value to error out. The error in cluster contains the following parameters:
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **instrument handle out** has the same value as the **instrument handle**.
- 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.

- **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 a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Acquisition:Acquisition Type

Short Name: Acquisition Type

Specifies how the digitizer acquires data and fills the waveform record.

Defined Values

Normal Flex Res DDC



Note Acquisition type DDC applies to the NI 5620/5621 only. To use DDC mode in the NI 5142, leave **acquisition type** set to Normal and set <u>DDC Enabled</u> to TRUE.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Acquisition: Advanced: Enable RIS in Auto Setup

Short Name: Enable RIS is Auto Setup

Indicates whether the digitizer should use RIS sample rates when searching for a frequency in autosetup.

Data Type	ViBoolean
Access	R/W
Channel Based	No

Acquisition:Binary Sample Width

Short Name: 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. 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 <u>Resolution</u>.

Defined Values

8, 16, 32

Data Type	Vilnt32
Access	R/W
Channel Based	No

Acquisition: Delay Before Initiate

Short Name: Delay Before Initiate

Specifies a delay in seconds that is used by <u>niScope_Initiate Acquisition</u> to allow additional delay between programming of the vertical range, trigger level, DDC, and the start of the acquisition. This property is supported only on the NI 5112 and the NI 5620/5621.

Default Value: 0.0

Data Type	ViReal64
Access	R/W
Channel Based	No

Acquisition:Resolution

Short Name: Resolution

Indicates the actual resolution in bits of valid data (as opposed to padding bits) in the acquired waveform. Compare to <u>Binary Sample Width</u>.

Valid Values: 8 to 32

Data Type	Vilnt32
Access	RO
Channel Based	No

Acquisition:Sample Mode

Short Name: Sample Mode

Returns the sample mode the digitizer is currently using.

Defined Values

Real Time Equivalent Time

Data Type	Vilnt32
Access	RO
Channel Based	No

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.

Default Value: " "

Data Type	ViString
Access	R/W
Channel Based	No

Clocking:Advanced:Oscillator Phase DAC Value

Short Name: 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. If this value is set, sample clock adjustments and TClk cannot do any subsample adjustment of the timebase sample clock.

Default Value: " "

The following table lists the characteristics of this property.

Data Type Vilnt32 Access R/W

Clocking:Clock Sync Pulse Source

Short Name: Clock Sync Pulse Source

For the NI 5102, specifies the line on which the sample clock is sent or received. For the NI 5112/5620/5621/5911, specifies the line on which the one-time sync pulse is sent or received. This line should be the same for all devices to be synchronized.

Defined Values

NO_SOURCE RTSI_0 RTSI_1 RTSI_2 RTSI_3 RTSI_4 RTSI_5 RTSI_6 PFI_1 PFI_2

Data Type	ViString
Access	R/W
Channel Based	No

Clocking:Exported Sample Clock Output Terminal

Short Name: Exported Sample Clock Output Terminal

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

CLK OUT RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PXI Star PFI 0 PFI 1 PFI 2 Clock Out

Data Type	ViString
Access	R/W
Channel Based	No

Clocking:Output Clock Source

Short Name: Output Clock Source

Specifies the output source for the 10 MHz clock to which another digitizer's sample clock can be phased-locked. The NI 5102 uses a 20 MHz system clock.

Defined Values

None RTSI Clock PFI 0 PFI 1 PFI 2 CLK OUT RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 3 RTSI 4 RTSI 5 RTSI 6

Data Type	ViString
Access	R/W
Channel Based	No

Clocking:PLL Lock Status

Short Name: PLL Locked

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.

Data Type	ViBoolean
Access	RO
Channel Based	No

Clocking:Reference (Input) Clock Source

Short Name: Reference Clock Source

Specifies the input source for the PLL reference clock to which the digitizer is phase-locked. For the the NI 5102, this is the source of the board clock. For other <u>Tradiitonal NI-DAQ (Legacy) devices</u> this is the 10 MHz clock. For <u>SMC-based devices</u> this is the 1–20 MHz clock.

Defined Values

External SMA Connector (5620/5621 Only) No Source RTSI_Clock PFI 0 PFI 1 PFI 2 PXI Clock CLK IN

Data Type	ViString
Access	R/W
Channel Based	No

Clocking:Reference Clock Rate

Short Name: Reference Clock Rate

If <u>Input Clock Source</u> is an external source, specifies the frequency of the input, or reference, clock, to which the internal sample clock timebase is synchronized. The frequency is in hertz. Refer to <u>Features Supported by</u> <u>Device</u> for valid values.

Data Type	ViReal64
Access	R/W
Channel Based	No

Clocking:Sample Clock Timebase Divisor

Short Name: Sample Clock Timebase Divisor

If <u>Sample Clock Timebase Source</u> is an external source, specifies the ratio between the sample clock timebase rate and the actual sample rate, which can be slower.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Clocking:Sample Clock Timebase Rate

Short Name: Sample Clock Timebase Rate

Specifies the frequency in hertz of the external clock used as the timebase source if <u>Sample Clock Timebase Source</u> is an external source.

Data Type	ViReal64
Access	R/W
Channel Based	No

Clocking:Sample Clock Timebase Source

Short Name: Sample Clock Timebase Source

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 <u>Min Sample Rate</u> (for internal sources) or the <u>Sample Clock Timebase</u> <u>Divisor</u> (for external sources).

Defined Values

CLK IN PXI Star PFI 0 PFI 1 No Source

Data Type	ViString
Access	R/W
Channel Based	No

Device:Temperature

Short Name: Temperature

Returns the temperature of the device in degrees Celsius from the onboard sensor.

Data Type	Vilnt32
Access	RO
Channel Based	No

Device:Serial Number

Short Name: Serial Number Returns the serial number of the device.

Data Type	ViString
Access	RO
Channel Based	No

Fetch:Data Transfer Block Size

Short Name: 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.

Data Type	Vilnt32
Access	RW
Channel Based	No

Fetch:Fetch Backlog

Short Name: Fetch Backlog

Specifies the number of points acquired that have not been fetched yet.

Data Type	ViReal64
Access	RO
Channel Based	No

Fetch: Fetch Meas Num Samples

Short Name: Fetch Meas Number of Samples

Determines the number of samples to fetch from a digitizer when performing a measurement. –1 means fetch all samples from the <u>Fetch</u> <u>Offset</u> property to the end of the current record.

Default Value: -1

Data Type	Vilnt32
Access	R/W
Channel Based	No

Fetch:Fetch Number of Records

Short Name: 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.

Default Value: -1

Data Type	Vilnt32
Access	R/W
Channel Based	No

Fetch:Fetch Offset

Short Name: Fetch Offset

Sets the offset in samples; the samples returned also depend on the <u>Fetch Relative To</u> property.

Default Value: 0

Valid Values: All integers

Data Type	Vilnt32
Access	R/W
Channel Based	No

Fetch:Fetch Record Number

Short Name: Fetch Record Number Sets the record to fetch. The record is from a channel you specify. Default Value: 0 Valid Values: Values greater than or equal to 0

Data Type	Vilnt32
Access	R/W
Channel Based	No

Fetch:Fetch Relative To

Short Name: 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.

Defined Values

Start—Fetch data starting at the first point sampled by the digitizer.

Trigger—Fetch at the first posttrigger sample.

Pretrigger—Fetches relative to the first pretrigger point requested with <u>niScope Configure Horizontal Timing</u>.

Now—Fetch data at the last sample acquired.

Read Pointer—The read pointer is set to zero when a new acquisition is initiated. After every fetch the read pointer is incremeted to be the sample after the last sample retrieved. Therefore, you can repeatedly fetch relative to the read pointer for a continuous acquisition program.

Default Value: Pretrigger

Data Type	Vilnt32
Access	R/W
Channel Based	No

Fetch:Points Done

Short Name: Points Done

Actual number of samples acquired since the last fetch, relative to the configured value for <u>Fetch Relative To</u>, including <u>Fetch Offset</u>, and for the current configured <u>Fetch Record Number</u>.

Data Type	ViReal64
Access	RO
Channel Based	No

Fetch:Records Done

Short Name: Records Done

Returns the number of records your digitizer has acquired.

Data Type	Vilnt32
Access	RO
Channel Based	No

Horizontal:Acquisition Start Time

Short Name: Acq Start Time

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).

Data Type	ViReal64
Access	RO
Channel Based	No

Horizontal:Actual Record Length

Short Name: 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 <u>niScope Configure Horizontal Timing</u>.

Valid Values: 1 to the maximum memory size

Data Type	Vilnt32
Access	RO
Channel Based	No

Horizontal:Actual Sample Rate

Short Name: Actual Sample Rate Returns the actual sample rate used for the acquisition. Units: hertz (Samples / Second)

Data Type	ViReal64
Access	RO
Channel Based	No

Horizontal:Advanced:5102 Adjust Pretrigger Samples

Short Name: Adjust Samples

When set to TRUE and the digitizer is set to master, the number of pretrigger samples and total number of samples are adjusted to enable synchronizing a master and slave NI 5102.

Data Type	ViBoolean
Access	R/W
Channel Based	No

Horizontal:Enable Records > Memory

Short Name: Enable Records > Memory

Allows you to acquire more records than fit in onboard memory. Only works in digitizers that support continuous acquisition. Refer to <u>Features</u> <u>Supported by Device</u> to find out if your digitizer supports continuous acquisition.

Defined Values

TRUE—Enables NI-SCOPE to fetch more records than fit in memory

FALSE—Disables NI-SCOPE to fetch more records than fit in memory

Data Type	ViBoolean
Access	R/W
Channel Based	No

Horizontal:Enable TDC

Short Name: Enable TDC

Specifies that the digitizer should record the trigger position precisely using time-digital conversion (TDC). Disabling TDC by setting this property to FALSE may reduce rearm time but causes the digitizer to round the trigger position to the nearest sample clock.

Data Type	ViBoolean
Access	R/W
Channel Based	No

Horizontal:Enable Time Interleaved Sampling

Short Name: Enable TIS

Extends the maximum sample rate on the specified Active 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: FALSE (0)

Defined Values

TRUE (1)—Use multiple interleaved ADCs to acquire data for this channel.

FALSE (0)—Use only this channel's ADC to acquire data for this channel.

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

Horizontal:Enforce Realtime

Short Name: Enforce Realtime

Indicates whether the digitizer enforces real-time measurements or allows equivalent-time measurements.

Defined Values

TRUE—Allow real-time acquisitions only

FALSE—Allow real-time and equivalent-time acquisitions

Data Type	ViBoolean
Access	R/W
Channel Based	No

Horizontal:Maximum RIS Rate

Short Name: Maximum RIS Rate Returns the maximum RIS sampling rate in hertz.

Data Type	ViReal64
Access	RO
Channel Based	No

Horizontal:Maximum Real Time Sample Rate

Short Name: Maximum Real Time Sample Rate Returns the maximum real-time sample rate in hertz.

Data Type	ViReal64
Access	RO
Channel Based	No

Horizontal:Memory Size

Short Name: Memory Size

Returns the total combined amount of onboard memory for all channels in bytes.

Data Type	Vilnt32
Access	RO
Channel Based	No

Horizontal: Min Number of Points

Short Name: Min Number of Points

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 <u>Horizontal Actual Record Length</u> property returns the actual record length.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Horizontal:Min Sample Rate

Short Name: Min Sample Rate

Specifies the sampling rate for the acquisition.

Units: samples per second

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 <u>niScope Configure Acquisition</u> and not require more memory than the onboard memory module allows.

Data Type	ViReal64
Access	R/W
Channel Based	No

Horizontal:RIS Method

Short Name: RIS Method

Specifies the algorithm for random-interleaved sampling, which is used if the sample rate exceeds the <u>Max Realtime Sampling Rate</u>.

Defined Values

Exact Num Avg (default mode) Min Num Avg Incomplete Limited Bin Width

Data Type	Vilnt32
Access	R/W
Channel Based	No

Horizontal:RIS Num Avg

Short Name: RIS Num Avg

Specifies the number of averages in each RIS bin. Averaging is useful in RIS since 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, which helps minimize the noise in the reconstructed waveform.

Valid Values: Greater than or equal to 0

Data Type	Vilnt32
Access	R/W
Channel Based	No

Horizontal:Reference Position

Short Name: Reference Position

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 <u>Trigger Delay</u> 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.

Data Type	ViReal64
Access	R/W
Channel Based	No

Horizontal:Time Per Record

Short Name: Time Per Record

Specifies the length of time that corresponds to the record length. The units are seconds.

Data Type	ViReal64
Access	R/W
Channel Based	No

IF Digitizer (562x):AGC:Average Control

Short Name: AGC Average Control

Averages the <u>AGC</u> values.

Default Value: Mean

Defined Values

Mean Median

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Loop Gain 0 Exponent

Short Name: AGC Loop Gain 0 Exponent Along with <u>AGC Loop Gain 0 Mantissa</u>, sets the loop gain for the <u>AGC</u>. Default Value: 0

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Loop Gain 0 Mantissa

Short Name: AGC Loop Gain 0 Mantissa Along with <u>AGC Loop Gain 0 Exponent</u>, sets the loop gain for the <u>AGC</u>. Default Value: 0

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Loop Gain 1 Exponent

Short Name: AGC Loop Gain 1 Exponent Along with <u>AGC Loop Gain 1 Mantissa</u>, sets the loop gain for the <u>AGC</u>. Default Value: 0

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Loop Gain 1 Mantissa

Short Name: AGC Loop Gain 1 Mantissa Along with <u>AGC Loop Gain 1 Exponent</u>, sets the loop gain for the <u>AGC</u>. Default Value: 0

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Lower Gain Limit

Short Name: AGC Lower Gain Limit Sets the minimum gain and maximum signal levels in the <u>AGC</u>. Default Value: 6.020600

Data Type	ViReal64
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Threshold

Short Name: AGC Threshold Sets the gain error in the <u>AGC</u>. Default Value: 0x034D

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):AGC:Upper Gain Limit

Short Name: AGC Upper Gain Limit Sets the maximum gain and minimum signal levels in the <u>AGC</u>. Default Value: 6.020600

Data Type	ViReal64
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Coordinate Converter Input

Selects the source for the input to the coordinate converter, either the HB filter or the programmable FIR.

Default Value: Programmable FIR

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:DDC Direct Register Address

Short Name: DDC Direct Register Address

Used for directly accessing the <u>DDC</u> registers.

Data Type	Vilnt32
Access	R/W
Channel Based	No

IF Digitizer (562x):Advanced:DDC Direct Register Data

Short Name: DDC Direct Register Data Used for directly accessing the <u>DDC</u> registers. Default Value: 0

Data Type	Vilnt32
Access	R/W
Channel Based	No

IF Digitizer (562x):Advanced:Device Number

Short Name: Device Number

Indicates the device number associated with the current session.

Data Type	Vilnt32
Access	RO
Channel Based	No

IF Digitizer (562x):Advanced:Enable DDC

Short Name: Enable DDC

Set this property to FALSE to disable programming the <u>DDC</u>.



Notes

- This attribute is supported for NI 5620/5621 digitizers only. For NI 514X digitizers, use <u>DDC Enabled</u>
- Custom programming of the DDC using NI-SCOPE property nodes is not supported by National Instruments.
- National Instruments supports using the DDC only when the Modulation Toolkit and/or Spectral Measurements Toolkit are used, because they make use of the DDC automatically (that is, without user intervention) when configuration settings allow.

Default Value: TRUE

Data Type	ViBoolean
Access	R/W
Channel Based	No

IF Digitizer (562x):Advanced:Enable Dither

Short Name: Enable Dither

Applies dither at the input of the ADC. Set this property to TRUE to enable dither.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Fetch Interleaved Data

Short Name: Fetch Interleaved Data

Short Name: Serial DAC Cal Voltage

Set to TRUE to retrieve one array with alternating values on the NI 5620/5621. This property can be used 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.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	No

IF Digitizer (562x):Advanced:Q Input to Coordinate Converter

Short Name: Q Input to Coord Converter Either enables or zeros out the Q input to coordinate converter. Default Value: I and Q

Defined Values

Q I and Q

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Resampler:Bypass

Short Name: Resampler Bypass

Either enables or bypasses the resampler filter in the DDC. The resampler is 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.

Default Value: TRUE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Resampler:Filter Mode

Short Name: Resampler Filter Mode Selects the resampling filter mode. Defined Values Resampler Enabled HB 1 Enabled Resampler and HB 1 Enabled Both HB Filters Enabled Resampler and Both HB Filters Enabled

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Resampler:NCO Divide

Short Name: Resampler NCO Divide

Divides down the <u>resampler</u> NCO output by the value loaded into the register plus one.

Default Value: 2

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Resampler:Output Pulse Delay

Short Name: Resampler Output Pulse Delay

Programs the delay between output samples when interpolating. These outputs can be delayed from 2 to 255 clocks.

Default Value: 16

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Resampler:Reference Divide

Short Name: Resampler Reference Divide

Divides down the reference clock by the value loaded into the register plus one. Load with a value that is one less than the desired period.

Default Value: 2

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Serial DAC Cal Voltage

Short Name: Serial DAC Cal Voltage

Specifies the voltage of the DAC that controls the oscillator; this property is used for external calibration.

Data Type	ViReal64
Access	R/W
Channel Based	No

IF Digitizer (562x):Advanced:Syncout CLK Select

Short Name: Syncout CLK Select Source for Syncout CLK. Default Value: CLK IN

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Test Mode Sin/Cos

Short Name: Test Mode Sin/Cos

Enables the special test mode where the carrier NCO outputs are set to 0x7FFF.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Timing NCO:Center Frequency

Short Name: Timing NCO Center Freq.

Controls the frequency of the timing NCO. Specifies the timing NCO center frequency in binary format:

 $N = (F_{out} / F_{resampler}) \& 2^{32}$

where F_{out} is the output frequency and $F_{resampler}$ is the resampled frequency.

The value is transferred to the active register during the next initiate acquisition operation.

Default Value: 0X800000

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Timing NCO:Clear Phase Accum.

Short Name: Timing NCO Clear Phase Accum.

If FALSE, enables the accumulator in the <u>timing NCO</u>. If TRUE, zeros out feedback in the accumulator.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

Timing NCO Enable Offset Freq.

Short Name: Timing NCO Enable Offset Freq.

If TRUE, enables offset frequency in the $\underline{\text{timing NCO}}$. If FALSE, applies no offset frequency.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Timing NCO:Frequency Offset Bits

Short Name: Timing NCO Freq. Offset Bits Specifies the number of offset bits in the <u>timing NCO</u>. Default Value: 8 bits

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Timing NCO:Phase Accum. Load on Update

Short Name: Timing NCO Phase Accum. Load on Update

When TRUE, updates the <u>timing NCO</u> frequency to zero the feedback of the phase accumulator as well as update the phase and frequency.

Default Value: TRUE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Advanced:Timing NCO:Phase Offset

Short Name: Timing NCO Phase Offset

Offsets the phase of the timing NCO in binary format.

The value is transferred to the active register during the next initiate acquisition.

Default Value: 0

Valid Range: 0 to 6.283185307179586476925286766558

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):CIC Filter:Decimation

Short Name: CIC Decimation

Controls the decimation in the CIC filter. The CIC filter reduces the sample rate of a wideband signal to a rate that other filters in the DDC can process.

Default Value: 4

Valid Range

4 to 32

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):CIC Filter:Shift Gain

Short Name: CIC Shift Gain

Controls the shift gain at the input to the CIC filter. The CIC filter reduces the sample rate of a wideband signal to a rate that other filters in the <u>DDC</u> can process.

Default Value: 0

Valid Range

0 to 15

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Carrier Filter: NCO Center Frequency

Short Name: Carrier NCO Center Frequency

Controls the frequency of the carrier NCO. The coerced value can be read back.

Default Value: 15.0e6

Data Type	ViReal64
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Carrier Mixer:Phase Offset

Short Name: Carrier Phase Offset

Offsets the phase of the carrier NCO. The coerced value can be read back.

Default Value: 0.0

Data Type	ViReal64
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Combined Decimation

Short Name: Combined Decimation Returns the combined <u>DDC</u> decimation.

Data Type	Vilnt32
Access	RO
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:Delay

Short Name: Discriminator Delay

Sets the number of delays in the discriminator.

Default Value: 1

Valid Range

1 to 8

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:Enable

Short Name: Discr. Enable

Enables or disables the discriminator. If set to TRUE, frequency discriminator is enabled.

Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:FIR Decimation

Short Name: Discriminator FIR Decimation Sets the amount of decimation, from 1 to 8.

Valid Range

1 to 8 Default Value: 1

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:FIR Input Source

Short Name: Discriminator FIR Input Source

Sets the discriminator FIR input source to phase, magnitude, or resampler.

Default Value: Phase

Defined Values

Phase (0)

Magnitude (1)

Resampler (2)

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:FIR Symmetry

Short Name: Discriminator FIR Symmetry

Sets the discriminator FIR symmetry to symmetric or asymmetric.

Default Value: Symmetric

Defined Values

Symmetric (0) Asymmetric (1)

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:FIR Symmetry Type

Short Name: Discriminator FIR Symmetry Type

Sets the discriminator FIR symmetry type to even or odd.

Default Value: Even

Defined Values

Even (0) Odd (1)

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:FIR Taps

Short Name: Discriminator FIR Taps

Sets the discriminator FIR number of taps.

Default Value: 1

Valid Range

1 to 63

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Frequency Discriminator:Phase Multiplier

Short Name: Discriminator Phase Multiplier

Programs the coordinate converter to multiply the phase output by 1, 2, 4, or 8. Multiplying the phase output removes phase modulation before the frequency is measured.

Default Value: 0

Valid Range

1 to 16

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Bypass

Short Name: Halfband Filter Bypass

Enables or bypasses the halfband filters. If set to TRUE, halfband filters are bypassed.

Default Value: TRUE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Filter 1 Enable

Short Name: Halfband Filter 1 Enable Enables halfband filter 1. If TRUE, filter is enabled. Default Value: TRUE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Filter 2 Enable

Short Name: Halfband Filter 2 Enable Enables halfband filter 2. If TRUE, filter is enabled. Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Filter 3 Enable

Short Name: Halfband Filter 3 Enable Enables halfband filter 3. If TRUE, filter is enabled. Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Filter 4 Enable

Short Name: Halfband Filter 4 Enable Enables halfband filter 4. If TRUE, filter is enabled. Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Halfband Filter:Filter 5 Enable

Short Name: Halfband Filter 5 Enable Enables halfband filter 5. If TRUE, filter is enabled. Default Value: FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Output Configuration:Parallel:AOUT Source

Short Name: AOUT Parallel Output Source

Specifies the source for the AOUT parallel output from the DDC.

Default Value: I Data

Defined Values

l Data Magnitude

Frequency

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Output Configuration:Parallel:BOUT Source

Short Name: BOUT Parallel Output Source
Specifies the source for the BOUT parallel output from the DDC.
Default Value: Q Data
Defined Values
Q Data
Phase Data

Magnitude Data

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Programmable FIR Filter:Decimation

Short Name: Prog. FIR Filter Decimation Specifies the programmable FIR filter decimation. Default Value: 1

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Programmable FIR Filter:Real/Complex

Short Name: Prog. FIR Filter Real/Complex
Sets either a complex filter or a dual real filter.
Default Value: Real
Defined Values
Real

Complex

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Programmable FIR Filter:Symmetry

Short Name: Prog. FIR Filter Symmetry Sets either a symmetric or asymmetric filter. Default Value: Symmetric

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Programmable FIR Filter:Symmetry Type

Short Name: Prog. FIR Filter Symmetry Type Sets either even or odd symmetry. Default Value: Even

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

IF Digitizer (562x):Programmable FIR Filter:Taps

Short Name: Prog. FIR Filter Taps Defines the number of taps (in other words, coefficients) for a FIR filter. Default Value: 25

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Inherent IVI Attributes:Advanced Session Information:Logical Name

Short Name: Logical Name

A string that contains the logical name you specified when opening the current IVI session. You can pass a logical name to <u>niScope Initialize</u> or <u>niScope Initialize With Options</u>. 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.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Advanced Session Information:Resource Descriptor

Short Name: 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.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Capabilities:Channel Count

Short Name: 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.

Data Type	Vilnt32
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Capabilities:Class Group Capabilities

Short Name: Class Group Capabilities

A string that contains a comma-separated list of class-extension groups that this driver implements.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Capabilities:Supported Instrument Models

Short Name: Supported Instrument Models

A string that contains a comma-separated list of the instrument model numbers supported by this driver.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Class Specification Major Version

Short Name: Class Specification Major Version

The major version number of the class specification with which this driver is compliant.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Class Specification Minor Version

Short Name: Class Specification Minor Version

The minor version number of the class specification with which this driver is compliant.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Description

Short Name: Description

A string that contains the description of the instrument.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Driver Prefix

Short Name: 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.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Driver Vendor

Short Name: Driver Vendor

A string that contains the name of the vendor that supplies this driver, for example, "National Instruments".

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Driver Identification:Revision

Short Name: Revision

The string that contains additional version information about this instrument driver.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Instrument Identification:Firmware Revision

Short Name: Firmware Revision

A string that contains the firmware revision information for the current instrument.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Instrument Identification:Manufacturer

Short Name: Manufacturer

A string that contains the name of the instrument manufacturer, for example, "National Instruments".

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:Instrument Identification:Model

Short Name: Model

A string that contains the model number of the current instrument.

Data Type	ViString
Access	RO
Channel Based	No

Inherent IVI Attributes:User Options:Cache

Short Name: Cache

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 <u>niScope Initialize with Options</u> to override this value.

Data Type	ViBoolean
Access	R/W
Channel Based	No

Inherent IVI Attributes:User Options:Interchange Check

Short Name: Interchange Check

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.

Defined Values

TRUE (1) FALSE (0)

Data Type	ViBoolean
Access	R/W
Channel Based	No

Inherent IVI Attributes:User Options:Query Instrument Status

Short Name: Query Instrument Status

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 <u>niScope Initialize with Options</u> to override this value.

Defined Values

TRUE (1) FALSE (0)

Data Type	ViBoolean
Access	R/W
Channel Based	No

Inherent IVI Attributes:User Options:Range Check

Short Name: 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 <u>niScope Initialize with Options</u> to override this value.

Defined Values

TRUE (1) FALSE (0)

Data Type	ViBoolean
Access	R/W
Channel Based	No

Inherent IVI Attributes:User Options:Record Value Coercions

Short Name: Record Value Coercions

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 <u>niScope Initialize with Options</u> to override this value.

Defined Values

TRUE (1) FALSE (0)

Data Type	ViBoolean
Access	R/W
Channel Based	No

Inherent IVI Attributes:User Options:Simulate

Short Name: Simulate

Specifies whether to simulate instrument driver I/O operations. The default value is FALSE. Use <u>niScope Initialize with Options</u> to override this value.

Defined Values

TRUE (1) FALSE (0)

Data Type	ViBoolean
Access	R/W
Channel Based	No

Synchronization:5 Volt Power:Output Terminal

Short Name: 5VoltPower.OutputTerm

Specifies the destination for the 5 Volt power signal. Refer to the device specifications document for a list of valid destinations.

Defined Values

- 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_PXI_STAR PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Advance Trigger:Output Terminal

Short Name: ExportedAdvTrig.OutputTerm

Specifies the destination for the advance trigger. When the advance trigger is received, the digitizer begins acquiring pretrigger samples.

Defined Values

Immediate RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PFI 0 PFI 1 PFI 2 PXI Star

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization: Advance Trigger: Source

Short Name: 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.

Defined Values

"VAL_IMMEDIATE"
"VAL_RTSI_0"
"VAL_RTSI_1"
"VAL_RTSI_2"
"VAL_RTSI_3"
"VAL_RTSI_4"
"VAL_RTSI_6"
"VAL_RTSI_6"
"VAL_PFI_0"
"VAL_PFI_1"
"VAL_PFI_2"
"VAL_PFI_2"
"VAL_PXI_STAR"
"VAL_SW_TRIG_FUNC"

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization: Arm Reference Trigger Source

Short Name: 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.

Defined Values

"VAL_IMMEDIATE"
"VAL_RTSI_0"
"VAL_RTSI_1"
"VAL_RTSI_2"
"VAL_RTSI_3"
"VAL_RTSI_4"
"VAL_RTSI_6"
"VAL_RTSI_6"
"VAL_PFI_0"
"VAL_PFI_1"
"VAL_PFI_2"
"VAL_PFI_2"
"VAL_PXI_STAR"
"VAL_SW_TRIG_FUNC"

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:End of Acquisition:Output Terminal

Short Name: EndofAcqEvent.OutputTerm

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 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_PXI_STAR PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization: End of Record: Output Terminal

Short Name: EndofRecEvent.OutputTerm

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_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_PXI_STAR PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Master Enable

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

Data Type	ViBoolean
Access	R/W
Channel Based	No

Synchronization:Ready for Advance:Output Terminal

Short Name: RdyForAdvEvent.OutputTerm

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

RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PFI 0 PFI 1 PFI 2 PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Ready for Reference:Output Terminal

Short Name: RdyForRefEvent.OutputTerm

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.

Defined Values

RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PFI 0 PFI 1 PFI 2 PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Ready for Start:Output Terminal

Short Name: RdyForStartEvent.OutputTerm

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 device-specific documentation in the *NI High-Speed Digitizers Help* for a list of valid destinations for your device.

Defined Values

RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PFI 0 PFI 1 PFI 2 PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Record Arm Source

Short Name: Record Arm Source

Specifies the source for the record arm.

Defined Values

"VAL_IMMEDIATE" "VAL_RTSI_0" "VAL_RTSI_1" "VAL_RTSI_2" "VAL_RTSI_3" "VAL_RTSI_3" "VAL_RTSI_4" "VAL_RTSI_5" "VAL_RTSI_6" "VAL_PFI_1" "VAL_PFI_2"

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Start Trigger (Acq. Arm):Output Terminal

Short Name: StartTrig.OutputTerm

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 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_PXI_STAR PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Start Trigger (Acq. Arm):Source

Short Name: Start Trigger 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.

Defined Values

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_PXI_STAR	PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Synchronization:Trigger Calibration Delay:Slave Trigger Delay

Short Name: Slave Trigger Delay

Specifies the delay in seconds for the trigger from the master to the slave. This value adjusts the **initial X** value of the slave digitizers to correct for the propagation delay between the master trigger output and slave trigger input.

Data Type	ViReal64
Access	R/W
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger from PFI Delay

Short Name: Trigger from PFI Delay

A factory-programmed value that specifies the delay in seconds for the PFI lines to the trigger input. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave digitizers, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> attribute.

Data Type	ViReal64
Access	RO
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger from RTSI Delay

Short Name: Trigger from RTSI Delay

A factory-programmed value that specifies the delay in seconds for the RTSI bus to the trigger input. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave digitizers, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> property.

Data Type	ViReal64
Access	RO
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger from Star Delay

Short Name: Trigger from Star Delay

A factory-programmed value that specifies the delay in seconds for PXI Star Trigger line to the trigger input. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave digitizers, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> property.

Data Type	ViReal64
Access	RO
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger to PFI Delay

Short Name: Trigger to PFI Delay

A factory-programmed value that specifies the delay in seconds for the trigger to the PFI lines. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave digitizers, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> property.

Data Type	ViReal64
Access	RO
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger to RTSI Delay

Short Name: Trigger to RTSI Delay

A factory-programmed value that specifies the delay in seconds for the trigger to the RTSI bus. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave digitizers, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> property.

Data Type	ViReal64
Access	RO
Channel Based	No

Synchronization:Trigger Calibration Delay:Trigger to Star Delay

Short Name: Trigger to Star Delay

A factory-programmed value that specifies the delay in seconds for the trigger to the PXI Star Trigger line.. By itself, this property has no effect on the acquired data. However, depending on how the trigger lines are routed between the master and slave boards, you can use this value as a starting point to set the <u>Slave Trigger Delay</u> property.

Data Type	ViReal64
Access	RO
Channel Based	No

Triggering: Trigger Auto Triggered

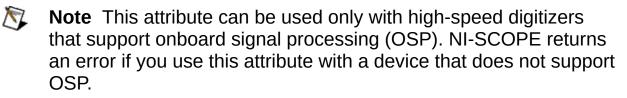
Short Name: Trigger Auto Triggered Specifies whether the acquisition was auto triggered.

Data Type	ViBoolean	
Access	RO	
Channel Based	Based No	

Triggering:Onboard Signal Processing:Ref Trigger Detector Location

Short Name: Ref Trigger Detector Location

Specifies which reference trigger detection circuitry to use on the device.



Default Value: Analog Detection Circuit

Valid Values

DDC Output—Use the onboard signal processing logic to implement the reference trigger. This option detects trigger conditions by analyzing the processed digital signal.

Analog Detection Circuit—Use the hardware analog circuitry to implement the reference trigger. This option detects trigger conditions by analyzing the unprocessed analog signal.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Onboard Signal Processing:Ref Trig Min Quiet Time

Short Name: Ref Trig 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 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.

The following table lists the characteristics of this property.

Data Type	ViReal64
Access	R/W
Channel Based	No

Related Topics

Burst Triggers

Triggering:Start to Ref Trigger Holdoff

Short Name: Start to Ref Trigger Holdoff

Pass 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

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Triggering: Trigger Coupling

Short Name: Trigger Coupling

Specifies how the digitizer couples the trigger source. This property affects instrument operation only when the <u>Trigger Type</u> 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.

Defined Values

AC DC HF Reject LF Reject AC Plus HF Reject

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Trigger Delay

Short Name: Trigger Delay

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.

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering:Trigger Holdoff

Short Name: 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. 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

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering: Trigger Hysteresis

Short Name: Trigger Hysteresis

Specifies 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 Trigger Level parameter, has the slope you specify with the Trigger Slope parameter, and passes through the hysteresis window that you specify with this parameter.

Units: Volts

Valid Values:

Min Value: 0

Max Value for positive trigger slope:

```
Hysteresis – trigger level >= –(vertical range/2) + vertical offset
```

Max value for negative trigger slope:

Hysteresis + trigger level <= (vertical range/2) + vertical offset

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering:Trigger Impedance

Short Name: Trigger Impedance Sets the impedance for the trigger channel (NI 5112 only) Default Value: $1 M\Omega$

Defined Values

1 MΩ 50 Ω

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering:Trigger Level

Short Name: Trigger Level

Specifies the voltage threshold for the trigger. The units are volts. This property affects instrument behavior only when the <u>Trigger Type</u> is set to Edge, Hysteresis, or Window.

Valid Range: The values of the **range** and **offset** parameters in <u>niScope</u> <u>Configure Vertical</u> 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 <= Vertical Range/2 + Vertical Offset Trigger Level >= (–Vertical Range/2) + Vertical Offset

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering: Trigger Modifier

Short Name: Trigger Modifier

Configures the device to automatically complete an acquisition if a trigger has not been received.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Ref Trigger Output Terminal

Short Name RefTrigger.OutputTerm

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	External ⁻	TRIG input
--------------	-----------------------	------------

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_PXI_STAR	PXI Star Trigger

Data Type	ViString
Access	R/W
Channel Based	No

Triggering:Trigger Slope

Short Name: Trigger Slope

Specifies whether a rising or a falling edge triggers the digitizer.

This property affects instrument operation only when the <u>Trigger Type</u> property is set to edge, hysteresis, window, or video.

Defined Values

Positive Negative

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering: Trigger Source

Short Name: Trigger Source

Specifies the source the digitizer monitors for the trigger event. The value must be selected from one of the defined values.

Defined Values

0 1 2 3 4 5 6 7 VAL EXTERNAL 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 PXI STAR VAL SW TRIG FUNC

Data Type	ViString
Access	R/W
Channel Based	No

Triggering:Trigger Type

Short Name: Trigger Type Specifies the type of trigger to use.

Defined Values

Edge Hysteresis Digital Window Immediate Software Video

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Trigger Video:Enable DC Restore

Short Name: Enable DC Restore

Restores the video-triggered data retrieved by the digitizer to the video signal's zero reference point.

Default Value

FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	No

Triggering:Trigger Video:Event

Short Name: Video Trigger Event

Specifies the event to trigger on.

Defined Values

- (1) Field1 (the odd field)—triggers on the first line of Field 1
- (2) Field2 (the even field)-triggers on the first line of Field 2
- (3) Any Field—triggers on the vertical synch of a random field
- (4) Any Line—triggers on the first line available
- (5) Line Number-triggers on the beginning of a specific line

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Trigger Video:Line Number

Short Name: Video Line Number

Specifies the line number to trigger on. This property is only used if the video trigger <u>Event</u> is set as Line Number. Valid values depend on the video signal format selected.

Signal Format	Line Numbers
M-NTSC, 480i, 480p	1 to 525
BG/PAL, SECAM, 576i, 576p	1 to 625
720р	1 to 750
1080i,1080p	1 to 1,125

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Trigger Video:Polarity

Short Name: Video Polarity

Specifies whether the video signal is positive or negative.

Defined Values

Positive Negative

Data Type	Vilnt32
Access	R/W
Channel Based	No

Triggering:Trigger Video:Signal Format

Short Name: Video Signal Format Specifies the video signal format to use.

Defined Values

```
M-NTSC-(1)
B/G-PAL—(2)
SECAM—(3)
M-PAL-(1001)
480i/59.94 fps-(1010)
480i/60 fps-(1011)
480p/59.94 Fps-(1015)
480p/60 Fps-(1016)
576i/50 fps-(1020)
576p/50 Fps-(1025)
720p/50 Fps-(1031)
720p/59.94 Fps-(1032)
720p/60 Fps-(1033)
1080i/50 fps-(1040)
1080i/59.94 fps-(1041)
1080i/60 fps-(1042)
1080p/24 Fps-(1045)
```

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Triggering:Trigger Window:Trigger Window High Level

Short Name: Trigger Window High Level

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 <u>Trigger Window Low Level</u> property and this property.

Valid Range: The values of the <u>Vertical Range</u> and <u>Vertical Offset</u> properties determine the valid range for the Trigger Window Low Level property on the channel you specify with the <u>Trigger Source</u>.

The value you pass for this parameter must meet the following conditions:

High Trigger Level <= Vertical Range/2 + Vertical Offset High Trigger Level >= (–Vertical Range/2) + Vertical Offset High Trigger Level > Low Trigger Level

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering:Trigger Window:Trigger Window Low Level

Short Name: Trigger Window Low Level

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 <u>Trigger Window High Level</u>.

Valid Range: The values of the <u>Vertical Range</u> and <u>Vertical Offset</u> properties determine the valid range for this property on the channel you specify with the <u>Trigger Source</u>.

The value you pass for this parameter must meet the following conditions:

Low Trigger Level <= Vertical Range/2 + Vertical Offset Low Trigger Level >= (–Vertical Range/2) + Vertical Offset Low Trigger Level < High Trigger Level

Data Type	ViReal64
Access	R/W
Channel Based	No

Triggering:Trigger Window:Trigger Window Mode

Short Name: Trigger Window Mode

Specifies whether you want a trigger to occur when the signal enters or leaves the window specified by <u>Trigger Window Low Level</u> or <u>Trigger</u> <u>Window High Level</u>.

Defined Values

Entering (0)—Trigger upon entering the window Leaving (1)—Trigger upon leaving the window

Data Type	Vilnt32
Access	R/W
Channel Based	No

Vertical:Channel Enabled

Short Name: Channel Enabled

Specifies whether the digitizer acquires a waveform for the channel.

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

Vertical:Channel Terminal Configuration

Short Name: Terminal Configuration

Specifies how the digitizer configures the channel terminal.

Defined Values

- 0—Single Ended
- 1—Unbalanced Differential
- 2—Differential

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Vertical:Input Impedance

Short Name: Input Impedance

Specifies the input impedance for the channel in ohms.

Defined Values

50 Ω 1 MΩ

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Maximum Input Frequency

Short Name: Max. 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.

Special Values:

(-1)—Full bandwidth

(0)—Device default

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Probe Attenuation

Short Name: 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.

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Vertical Coupling

Short Name: Vertical Coupling

Specifies how the digitizer couples the input signal for the channel.

When changing input coupling, the input stage takes a finite amount of time to settle.

Defined Values

AC DC GND

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Vertical:Vertical Offset

Short Name: 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.

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Vertical Range

Short Name: 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 the Vertical Range property to 10.0 V.

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Advanced:Flex FIR Antialias Filter Type

Short Name: 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.

Default Value: 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.

48 Tap This filter is optimized for alias protection and frequency-Standard domain flatness.

(0)

- 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: 0.4 × sample rate
- Flatness: Ripple ranges from 0.005 dB to 0.120 dB depending on selected sample rate (refer to the *NI PXI-5922 Specifications* for more information)
- Rise time: Approximately 0.75/(sample rate)

48 Tap This filter is optimized for the lowest possible bandwidth for aHanning 48 tap filter and maximizes the signal-to-noise ratio (SNR).

- Settling time: within 14 samples from a 50% vertical trigger point
- Cutoff frequency: 0.030 × sample rate
- Flatness: 0 to –3 dB within cutoff frequency
- Rise time: 11.6/(sample rate)

16 Tap This filter is optimized for the lowest possible bandwidth for a Hanning 16 tap filter and maximizes the SNR.

(2)

(1)

- Settling time: 6 samples from a 50% vertical trigger point
- Cutoff frequency: 0.08 × sample rate
- Flatness: 0 to –3 dB within cutoff frequency
- Rise time: 7.7/(sample rate)

8 Tap This filter is optimized for the lowest possible bandwidth for an Hanning 8 tap filter and maximizes the SNR.

(3) • Settling time: 4 samples from a 50% vertical trigger

point

- Cutoff frequency: 0.15 × sample rate
- Flatness: 0 to –3 dB within cutoff frequency
- Rise time: 3.9/(sample rate)

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Vertical:Advanced:Digital Gain

Short Name: Digital Gain

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 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.

Units: Unitless

Default Value: 1

Valid Values: -1.5 to 1.5

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Vertical:Advanced:Digital Offset

Short Name: Digital Offset

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 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.

Units: Volts

Default Value: 0

Valid Values

 \pm (Vertical Range \times 0.4)

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC:Center Frequency

Short Name: Center Frequency

The frequency at which the $\underline{\text{DDC}}$ block frequency translates the input data.

 $\overline{\mathbb{N}}$

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

 $0 - (0.5 \times \text{Sample Clock Timebase Rate for digitizer})$

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC:Data Processing Mode

Short Name: Data Processing Mode

The way in which data is processed by the <u>DDC</u> block.

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.

Default Value: Complex

Defined Values

Real—The waveform data points are real numbers (I data)

Complex—The waveform data points are complex numbers (IQ data)

Data Type	Vilnt32
Access	R/W
Channel Based	No

Onboard Signal Processing (5142):DDC:DDC Enabled

Short Name: 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.



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. For NI 5620/5621 digitizers, use <u>Enable DDC</u>.

Default Value: FALSE

Defined Values

TRUE

FALSE

The following table lists the characteristics of this property.

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

✔ 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 × Sample Clock Timebase Rate). When using internal clocking, the Sample Clock Timbase Rate is 100 MS/s.

Related Topics

DDC Enabled Overview

Onboard Signal Processing (5142):DDC:Fetch Interleaved IQ Data

Short Name: Fetch Interleaved IQ Data

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: TRUE

Defined Values

VI_TRUE—A scalar fetch returns an array of waveforms in the following format: IQIQIQ...

VI_FALSE—A scalar fetch returns an array of waveforms in the following format: III...QQQ...

Data Type	Boolean
Access	R/W
Channel Based	No

Onboard Signal Processing (5142):DDC:Frequency Translation Enabled

Short Name: Frequency Translation Enabled

Enables/disables frequency translating the data around the user-selected center frequency down to baseband.



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.

Default Value: True

Defined Values

True

False

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC:Q Source

Short Name: Q Source

Specifies the channel that is the input to the Q data stream of the DDC.

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.

Default Value: The channel that the attribute is registered to

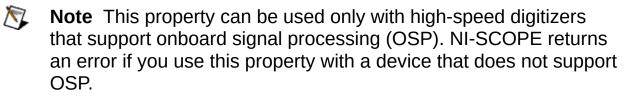
Valid Values: All valid channels for the device

Data Type	ViString
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC: IQ Signal Adjustments:Frequency Translation Phase:Frequency Translation Phase I

Short Name: Frequency Translation Phase I

The I oscillator phase in degrees at the first point acquired.



Units: Degrees

Default Value: 0

Valid Values

-360 to 360

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC: IQ Signal Adjustments:Frequency Translation Phase:Frequency Translation Phase Q

Short Name: Frequency Translation Phase Q

The Q oscillator phase in degrees at the first point acquired. Use this property only when the <u>Data Processing Mode</u> property is set to 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.

Units: Degrees

Default Value: 90

Valid Values

-360 to 360

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC:Signal Adjustments:Equalization:Equalization Filter Enabled

Short Name: 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.



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.

Default Value: FALSE

Defined Values

TRUE

FALSE

Data Type	ViBoolean
Access	R/W
Channel Based	Yes

Onboard Signal Processing (5142):DDC:Signal Adjustments:Equalization:Equalization Num Coefficients

Short Name: 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.



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.

Data Type	Vilnt32
Access	RO
Channel Based	Yes

Onboard Signal Processing (5142):OSP Overflow Error Reporting

Short Name: 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.

Valid Values

Default Value: Warning

Valid Values

Warning—Execution continues and NI-SCOPE returns a warning when an overflow has occurred in the OSP block.

Error—Execution stops and NI-SCOPE returns an error when an overflow has occurred in the OSP block.

Disabled—NI-SCOPE does not return an error when an overflow has occurred in the OSP block.

Data Type	Vilnt32
Access	R/W
Channel Based	No

Waveform Measurement: Array Gain

Short Name: Array Gain

Every element of an array is multiplied by this scalar value during the array gain measurement.

Default Value: 1.0

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement: Array Offset

Short Name: Array Offset

Every element of an array is added to this scalar value during the array offset measurement.

Default Value: 0.0

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement: Filter: Center Frequency

Short Name: Filter Center Frequency

The center frequency in hertz for filters of type bandpass and bandstop. The width of the filter is specified by <u>Filter Width</u>, where the cutoff frequencies are the center width.

Default Value: 1.0e6 Hz

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Filter:Cutoff Frequency

Short Name: 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.

Default Value: 1.0e6 Hz

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Filter:FIR Taps

Short Name: Filter 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.

Default Value: 25

Valid Values: >0

Data Type	Vilnt32
Access	R/W
Channel Based	No

Waveform Measurement:Filter:FIR Window

Short Name: Fir Filter Window

Specifies the FIR window type. The possible choices are:

None (0) Hanning Window (1) Hamming Window (2) Triangle_Window (3) Flat Top Window (4) Blackman Window (5)

The symmetric windows are applied to the FIR filter coefficients to limit passband ripple in FIR filters.

Default Value: 0-None

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Waveform Measurement:Filter:IIR Order

Short Name: Filter Order

Specifies the order of the infinite impulse response filter.

Default Value: 2

Valid Values: >0

Data Type	Vilnt32
Access	R/W
Channel Based	No

Waveform Measurement:Filter:Percent Waveform Transient

Short Name: Percent Waveform Transient

The percentage (0–100%) of the infinite impulse response (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%

Valid Range: 0.0–100.0%

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Filter:Ripple

Short Name: Filter Ripple

Specifies the amount of passband ripple for <u>Chebyshev filters</u>. More ripple gives a sharper cutoff for a given filter order.

Units: dB

Default Value: 0.1

Valid Values: >0.0

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement:Filter:Type

Short Name: Filter Type Specifies the type of digital filter.

Default Value: lowpass

Defined Values

- 0—lowpass
- 1—highpass
- 2—bandpass
- 3—bandstop

Data Type	Vilnt32
Access	R/W
Channel Based	No

Waveform Measurement:Filter:Width

Short Name: Filter Width

Specifies the width of a bandpass or bandstop filter. The cutoff frequencies are the (center frequency attribute $\pm 0.5 \times$ filter width).

Units: Hz

Default Value: 1.0e3

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement: Hysteresis Percent

Short Name: 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.

Default Value: 2%

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Interpolation:Polynomial Interpolation Order

Short Name: Poly. Interp. Order

Specifies the order of the polynomial used during the <u>polynomial</u> <u>interpolation</u> array 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 (linear interpolation)

Data Type	Vilnt32
Access	R/W
Channel Based	No

Waveform Measurement:Interpolation:Sampling Factor

Short Name: Interp. Sampling Factor

The new number of points for polynomial interpolation is the sampling factor times the input number of points. For example, if you acquire 1,000 points with the digitizer and set this property to 2.5, calling <u>niScope Fetch</u> <u>Measurement</u> (in the poly VI, select the Measurement Scalar DBL instance), with the <u>Polynomial Interpolation</u> measurement resamples the waveform to 2,500 points.

Default Value: 0.0

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Last Acq. Histogram Size

Short Name: 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 <u>voltage low</u> and <u>voltage high</u>.

Default Value: 256

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Waveform Measurement:Other Channel

Short Name: Other Channel

Specifies the second channel for two-channel measurements, such as <u>Add Channels</u>. If processing steps are registered with this channel, the processing happens before the waveform is used in a two-channel measurement.

Default Value: 0

Data Type	ViString
Access	R/W
Channel Based	Yes

Waveform Measurement:Reference Levels:Channel Based High Ref. Level

Short Name: Chan Based High Ref

Specifies the high reference level used in the measurements.

Units: Percentage or Volts based on the <u>Reference Level Units</u> property. Default Value: 90.0%

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Reference Levels:Channel Based Low Ref. Level

Short Name: Chan Based Low Ref

Specifies the low reference level used in the measurements.

Units: Percentage or Volts based on the <u>Reference Level Units</u> property. Default Value: 10.0%

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Reference Levels:Channel Based Mid Ref. Level

Short Name: Chan Based Mid Ref

Specifies the mid reference level used in the measurements.

Units: Percentage or Volts based on the <u>Reference Level Units</u> property. Default Value: 5.0%

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Reference Levels:Percentage Units Method

Short Name: Percentage Method

Specifies the method used to map percentage reference units to voltages.

Default Value: BaseTop

Defined Values

MinMax BaseTop LowHigh

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Waveform Measurement:Reference Levels:Units

Short Name: Ref. Level Units

Specifies the units for the waveform measurement reference levels. If you choose percentage, the measurement routine uses the <u>Percentage</u> <u>Units Method</u> property 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

Volts (0)

Percentage (1)

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Waveform Measurement:Time Histogram:High Time

Short Name: Time Hist. High Time

Specifies the maximum time limit 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. This value is used during the first time histogram measurement, and it is not updated until you call <u>niScope Clear</u> <u>Waveform Measurement Stats</u>.

Units: Seconds

Default Value: 5.0e-4

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement:Time Histogram:High Volts

Short Name: Time Hist. 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. This value is used during the first time histogram measurement, and it is not updated until you call <u>niScope</u> <u>Clear Waveform Measurement Stats</u>.

Default Value: 10.0 V

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Time Histogram:Low Time

Short Name: Time Hist. Low Time

Specifies the minimum time limit 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. This value is used during the first time histogram measurement, and it is not updated until you call <u>niScope Clear</u> <u>Waveform Measurement Stats</u>.

Units: Seconds

Default Value: -5.0e-4

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement:Time Histogram:Low Volts

Short Name: Time Hist. 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. This value is used during the first running time histogram measurement, and it is not updated until you call <u>niScope</u> <u>Clear Waveform Measurement Stats</u>.

Default Value: -10.0 V

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Waveform Measurement:Time Histogram:Size

Short Name: Time Hist. 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 the VI <u>niScope Clear Waveform Measurement</u> <u>Stats</u>.

Default Value: 256

Data Type	Vilnt32
Access	R/W
Channel Based	Yes

Waveform Measurement:Voltage Histogram:High Volts

Short Name: V. Hist. High Volts

Specifies the maximum voltage value in the running voltage histogram. This value is used during the first running voltage histogram measurement, and it is not updated until you call <u>niScope Clear</u> <u>Waveform Measurement Stats</u>.

Units: Volts

Default Value: 10.0

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement:Voltage Histogram:Low Volts

Short Name: V. Hist. Low Volts

Specifies the minimum voltage value in the running voltage histogram. This value is used during the first running voltage histogram measurement, and it is not updated until you call <u>niScope Clear</u> <u>Waveform Measurement Stats</u>.

Units: Volts

Default Value: -10.0

Data Type	ViReal64
Access	R/W
Channel Based	No

Waveform Measurement:Voltage Histogram:Size

Short Name: V. Hist. Size

Specifies the number of bins in the running voltage histogram. This value is used during the first running voltage histogram measurement, and it is not updated until you call <u>niScope Clear Waveform Measurement Stats</u>.

Default Value: 256

Data Type	Vilnt32
Access	R/W
Channel Based	No

Bandwidth

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 for this property are hertz.

This is an obsolete property. It is recommended that you use <u>Max Input</u> <u>Frequency</u> instead.

Data Type	ViReal64
Access	R/W
Channel Based	Yes

Trigger Output Event

Specifies the condition in which this device generates a digital pulse. Valid values depend on your digitizer. Refer to <u>Features Supported by</u> <u>Device</u>.

This property is either obsolete or is included for IVI compliance only. Consider using <u>niScope Export Signal</u> instead.

Defined Values

No Event Stop Trigger Reference Trigger Done

Data Type	Vilnt32
Access	R/W
Channel Based	No

Trigger Output Source

Specifies the hardware signal line on which the digital pulse is generated.

This property is either obsolete or is included for IVI compliance only. Consider using <u>niScope Export Signal</u> instead.

Defined Values

RTSI 0 RTSI 1 RTSI 2 RTSI 3 RTSI 4 RTSI 5 RTSI 6 PFI 0 PFI 1 PFI 2 PXI Star

Data Type	ViString
Access	R/W
Channel Based	No

Property Default Values

Expand this book or click one of the following links to view attribute default values for each device.

NI 5102 Property Default Values

NI 5112 Property Default Values

NI 5620/5621 Property Default Values

NI 5911 Property Default Values

NI SMC-Based Digitizers Property Default Values

SMC-Based Digitizers Property Default Values

The following table lists the property default values for NI SMC-based digitizers. N/A indicates that the attribute is not supported by the device.

Property (Attribute)	Default Value
Acq Start Time	0
Acquisition Type	Normal
Actual Record Length	1000
<u>Actual Sample</u> <u>Rate</u>	1000000
<u>Advance Trigger</u> <u>Source</u>	No Source
Arm Reference Trigger Source	No Source
<u>Array Gain</u>	1
Array Offset	0
Backlog	0
<u>Binary Sample</u> <u>Width</u>	NI 5105: 16 NI 5114: 8 NI 5122: 16 NI 5124: 16 NI 5142: 16 NI 5152: 8 NI 5922: 32
<u>Center Frequency</u>	NI 5105: N/A NI 5114: N/A NI 5122: N/A NI 5124: N/A NI 5142: 10 MHz NI 5152: N/A NI 5922: N/A
<u>Channel Based</u> High Ref	90

Channel Based	10
Low Ref	
<u>Channel Based</u> <u>Mid Ref</u>	50
Channel Count	NI 5105: 8
	NI 5114: 2
	NI 5122: 2
	NI 5124: 2
	NI 5142: 2
	NI 5152: 2
	NI 5922: 2
Channel Enabled	0
<u>Class Group</u>	IviScopeBase, IviScopeWaveformMeasurement,
Capabilities	IviScopeSampleMode, IviScopeAutoSetup,
	IviScopeTVTrigger
Clock Sync Pulse	No Source
Source	
Combined	N/A
Decimation	
Data Processing	NI 5105: N/A
Mode	NI 5114: N/A
<u>ivioue</u>	NI 5122: N/A
	NI 5124: N/A
	NI 5142: Complex
	NI 5152: N/A
	NI 5922: N/A
DDC Enchlad	
DDC Enabled	NI 5105: N/A NI 5114: N/A
	NI 5114. N/A NI 5122: N/A
	NI 5122: N/A NI 5124: N/A
	NI 5124: N/A NI 5142: FALSE
	NI 5152: N/A
	NI 5922: N/A
Delay Before	0
<u>Initiate</u>	

Description	National Instruments High Speed Digitizers Oscilloscope Instrument Driver
Digital Gain	NI 5105: N/A
	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: 1
	NI 5152: N/A
	NI 5922: N/A
Digital Offset	NI 5105: N/A
	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: 0
	NI 5152: N/A
	NI 5922: N/A
Driver Prefix	niScope
Driver Vendor	National Instruments
Enable DC	0
Restore	
Enable DDC	N/A
Enable Dither	N/A
Enable Records	0
<u>Greater Than</u>	
Memory	
Enable TDC	0
Enable TIS	NI 5105: N/A
	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: N/A
	NI 5152: FALSE
	NI 5922: N/A
Enforce Realtime	1
Equalization Filter	NI 5105: N/A

<u>Enabled</u>	NI 5114: N/A NI 5122: N/A	
	NI 5124: N/A	
	NI 5142: FALSE	
	NI 5152: N/A	
	NI 5922: N/A	
Equalization Num	NI 5105: N/A	
<u>Coefficients</u>	NI 5114: N/A	
	NI 5122: N/A	
	NI 5124: N/A	
	NI 5142: No Default	
	NI 5152: N/A	
	NI 5922: N/A	
Event	Any Line	
<u>Fetch Interleaved</u> Data	0	
Fetch Interleaved	NI 5105: N/A	
IQ Data	NI 5114: N/A	
	NI 5122: N/A	
	NI 5124: N/A	
	NI 5142: TRUE	
	NI 5152: N/A	
	NI 5922: N/A	
Fetch Meas Num Samples	-1	
Fetch Num	-1	
Records		
Fetch Offset	0	
Fetch Record	0	
Number		
Fetch Relative To	Pretrigger	
Filter Center	1000000	
Frequency		
Filter Cutoff	1000000	
Frequency		

Filter IIR Order	2	
Filter Ripple	0.1	
Filter Taps	25	
Filter Type	Lowpass	
Filter Width	1000	
FIR Filter Window	0	
FIR Filter Window	0	
Frequency Translation Enabled	NI 5105: N/A NI 5114: N/A NI 5122: N/A NI 5124: N/A	
	NI 5142: TRUE NI 5152: N/A NI 5922: N/A	
<u>Frequency</u> <u>Translation Phase</u> <u>I</u>	NI 5105: N/A NI 5114: N/A NI 5122: N/A NI 5124: N/A NI 5142: 0 NI 5152: N/A NI 5922: N/A	
<u>Frequency</u> <u>Translation Phase</u> Q	NI 5105: N/A NI 5114: N/A NI 5122: N/A NI 5124: N/A NI 5142: 90 NI 5152: N/A NI 5922: N/A	
<u>Hysteresis</u> Percent	2	
Input Clock Source	No Source	
Input Impedance	1000000	
Interchange	0	

<u>Interp. Sampling</u> <u>Factor</u>	2
Last ACQ Histogram Size	256
Line Number	0
Logical Name	
<u>Manufacturer</u>	National Instruments
Master Enable	0
<u>Max Input</u> <u>Frequency</u>	0, the driver selects the default bandwidth NI 5105: 24 MHz NI 5114: 125 MHz NI 5122: 35 MHz NI 5124: 60 MHz NI 5142: 35 MHz NI 5152: 300 MHz NI 5922: varies based on sample rate
<u>Max Realtime</u> <u>Sampling Rate</u>	NI 5105: 60 MS/s NI 5114: 250 MS/s NI 5122: 100 MS/s NI 5124: 200 MS/s NI 5142: 100 MS/s NI 5152: 1 GS/s NI 5922: 15 MS/s
<u>Max RIS Rate</u>	NI 5105: N/A NI 5114: 5 GS/S NI 5122: 2 GS/s NI 5124: 4 GS/s NI 5142: 2 GS/s NI 5152: 20 GS/s NI 5922: N/A
Min Number of Points	1000
Min Sample Rate	100000

Other Channel	0
Output Clock Source	No Source
OSP Overflow	NI 5105: N/A
Error Reporting	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: Warning
	NI 5152: N/A
	NI 5922: N/A
<u>Percentage</u> <u>Method</u>	Base Top
Percent	20
<u>Waveform</u>	
<u>Transient</u>	
PLL Locked	0
Points Done	0
<u>Polarity</u>	Negative
Poly. Interp. Order	1
Probe Attenuation	1
Prog FIR Filter	N/A
Taps	
<u>Q Source</u>	NI 5105: N/A
	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: The channel that the attribute is registered
	to
	NI 5152: N/A
	NI 5922: N/A
<u>Record Arm</u> <u>Source</u>	No Source
Records Done	0

Ref Trigger	NI 5105: N/A
Detector Location	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: Analog Detection Circuit
	NI 5152: N/A
	NI 5922: N/A
Ref Trig Min Quiet	NI 5105: N/A
<u>Time</u>	NI 5114: N/A
	NI 5122: N/A
	NI 5124: N/A
	NI 5142: 0
	NI 5152: N/A
	NI 5922: N/A
<u>Reference Clock</u> <u>Rate</u>	10 MHz
Reference Level	Percentage
<u>Units</u>	
<u>Reference</u>	50
<u>Position</u>	
Resolution	NI 5105: 12
	NI 5114: 8
	NI 5122: 14
	NI 5124: 12
	NI 5142: 14
	NI 5152: 8
	NI 5922: 22
<u>Resource</u>	device/session-specific
<u>Descriptor</u>	
RIS Method	Exact Num Avg
RIS Num Avg	4
Sample Clock	None
Output Terminal	
	1

Sample Clock	NI 5105: 60 MHz	
Timebase Rate	NI 5114: 250 MHz	
	NI 5122: 100 MHz	
	NI 5124: 200 MHz	
	NI 5142: 100 MHz	
	NI 5152: 1 GHz	
	NI 5922: 15 MHz	
Sample Clock Timebase Source	No Source	
Sample Mode	Real-Time	
Signal Format	NTSC	
<u>Slave Trigger</u> <u>Delay</u>	0	
Start Trigger	Immediate	
(Acq. Arm) <u>Source</u>		
<u>Supported</u>	—	
Instrument		
Models		
SYNCOUT Clock	N/A	
<u>Select</u>		
<u>Temperature</u>	N/A	
<u>Test Mode</u> <u>Sin/Cos</u>	N/A	
<u>Time Hist. High</u> <u>Time</u>	0.0005	
Time Hist. High	10	
<u>Volts</u>	0.0005	
<u>Time Hist. Low</u> <u>Time</u>	-0.0005	
<u>Time Hist. Low</u>	-10	
<u>Volts</u>		

<u>Time Hist. Size</u>	256	
Time Per Record	0.001	
Trigger Coupling	DC	
Trigger Delay	0	
Trigger From PFI Delay	0	
Trigger From RTSI Delay	0	
<u>Trigger From Star</u> Delay	0	
Trigger Holdoff	0	
Trigger Hysteresis	0.25	
<u>Trigger</u> Impedance	1000000	
Trigger Level	0	
<u>Trigger Output</u> Event	None	
Trigger Output Source	No Source	
Trigger Slope	Positive	
Trigger Source	No Source	
Trigger To PFI Delay	0	
Trigger To RTSI Delay	0	
<u>Trigger To Star</u> Delay	0	
Trigger Type	Immediate	
Trigger Window High Level	0	
Trigger Window Low Level	0	

<u>Trigger Window</u> <u>Mode</u>	Entering
Vertical Coupling	DC
Vertical Offset	0
Vertical Range	10.0
V. Hist. High Volts	10
V. Hist. Low Volts	-10
<u>V. Hist. Size</u>	256

NI 5102 Properties and Default Values

The following table lists the property default values for NI 5102 digitizers. N/A indicates that the attribute is not supported by the device.

Property (Attribute)	Default Value
Acquisition Arm Source (Start Trigger Source)	N/A
Acquisition Start Time	0
Acquisition Type	Normal
<u>Actual Record</u> Length	1000
Actual Sample Rate	100000
<u>Advance Trigger</u> <u>Source</u>	N/A
<u>Arm Reference</u> <u>Trigger Source</u>	N/A
Backlog	0
<u>Binary Sample</u> <u>Width</u>	8
Channel Count	2
Channel Enabled	0
<u>Class Group</u> <u>Capabilities</u>	IviScopeBase, IviScopeWaveformMeasurement, IviScopeSampleMode, IviScopeAutoSetup, IviScopeTVTrigger
<u>Clock Sync Pulse</u> <u>Source</u>	No Source
Combined Decimation	N/A
Delay Before Initiate	N/A
Description	National Instruments High Speed Digitizers

	Oscilloscope Instrument Driver
Driver Prefix	niScope
river Vendor	National Instruments
nable DC Restore	N/A
nable DDC	N/A
nable Dither	N/A
nable Records reater Than lemory	N/A
Enable TDC	N/A
nforce Realtime	1
vent	N/A
etch Interleaved ata	N/A
etch Meas Num amples	N/A
etch Num Records	N/A
etch Offset	N/A
tch Record mber	N/A
etch Relative To	Pretrigger
orizontal Min umber of Points	1000
put Clock Source	No Source
out Impedance	1000000
terchange Check	0
ne Number	N/A
gical Name	N/A
anufacturer	National Instruments
aster Enable	0
ax Input	1500000

Max Realtime	2000000	
Sampling Rate		
Max RIS Rate	N/A	
<u>Measurement Array</u> <u>Gain</u>	1	
<u>Measurement Array</u> <u>Offset</u>	0	
<u>Channel Based</u> High Ref	90	
<u>Channel Based Low</u> <u>Ref</u>	10	
<u>Channel Based Mid</u> Ref	50	
Filter Center Frequency	1000000	
Filter Cutoff Frequency	1000000	
Filter IIR Order	2	
Filter Ripple	0.1	
Filter Taps	25	
Percent Waveform Transient	20	
Filter Type	Lowpass	
Filter Width	1000	
Filter Window	0	
Hysteresis Percent	2	
Interpolation Sampling Factor	2	
Last ACQ Histogram Size	256	
Other Channel	0	

Polynomial Interpolation Order	1
Time Histogram High Time	0.0005
<u>Time Histogram</u> High Volts	10
<u>Time Histogram Low</u> <u>Time</u>	-0.0005
Time Histogram Low Volts	-10
Time Histogram Size	256
<u>Voltage Histogram</u> High Volts	10
<u>Voltage Histogram</u> Low Volts	-10
<u>Voltage Histogram</u> <u>Size</u>	256
<u>Min Sample Rate</u>	1000000
Model	NI <platform>-5102</platform>
Channel Count	2
Output Clock Source	No Source
Percentage Units Method	Base Top
PLL Locked	N/A
Points Done	1000
Polarity	N/A
Probe Attenuation	1
Record Arm Source	N/A
Records Done	1
Reference Clock	N/A

<u>Reference Level</u> <u>Units</u>	Percentage	
Reference Position	50	
Resolution	8	
Resource Descriptor	device/session-specific	
RIS Method	Exact Num Avg	
RIS Num Avg	1	
Sample Clock Output Terminal	N/A	
Sample Clock Timebase Divisor	N/A	
Sample Clock Timebase Rate	N/A	
Sample Clock Timebase Source	N/A	
Sample Mode	Real Time	
Signal Format	N/A	
Slave Trigger Delay	N/A	
SYNCOUT Clock Select	N/A	
<u>Temperature</u>	N/A	
Test Mode Sin/Cos	N/A	
Time Per Record	0.001	
Trigger Coupling	DC	
Trigger Delay	0	
Trigger From PFI Delay	N/A	
Trigger From RTSI Delay	N/A	
<u>Trigger From Star</u> Delay	N/A	

Trigger Holdoff	0
Trigger Hysteresis	0.25
Trigger Impedance	N/A
Trigger Level	0
<u>Trigger Output</u> <u>Event</u>	0
<u>Trigger Output</u> <u>Source</u>	No Source
Trigger Slope	Positive
Trigger Source	Immediate
Trigger To PFI Delay	N/A
<u>Trigger To RTSI</u> <u>Delay</u>	N/A
<u>Trigger To Star</u> <u>Delay</u>	N/A
Trigger Type	Immediate
<u>Trigger Window</u> <u>High Level</u>	0
<u>Trigger Window Low</u> <u>Level</u>	0
<u>Trigger Window</u> <u>Mode</u>	Entering
Vertical Coupling	DC
Vertical Offset	0
Vertical Range	10

NI 5112 Properties and Default Values

The following table lists the property default values for NI 5112 digitizers. N/A indicates that the attribute is not supported by the device.

Property (Attribute)	Default Value
<u>Acquisition Arm</u> <u>Source (Start</u> <u>Trigger Source)</u>	Immediate
<u>Acquisition Start</u> <u>Time</u>	0
Acquisition Type	Normal
<u>Actual Record</u> Length	1000
Actual Sample Rate	100000
<u>Advance Trigger</u> <u>Source</u>	N/A
<u>Arm Reference</u> <u>Trigger Source</u>	N/A
Backlog	0
<u>Binary Sample</u> <u>Width</u>	8
Channel Count	2
Channel Enabled	0
<u>Class Group</u> <u>Capabilities</u>	IviScopeBase, IviScopeWaveformMeasurement, IviScopeSampleMode, IviScopeAutoSetup, IviScopeTVTrigger
<u>Clock Sync Pulse</u> <u>Source</u>	No Source
Combined Decimation	N/A
Delay Before Initiate	0
Description	National Instruments High Speed Digitizers

	Oscilloscope Instrument Driver
Driver Prefix	niScope
Driver Vendor	National Instruments
Enable DC Restore	N/A
Enable DDC	N/A
Enable Dither	N/A
Enable Records Greater Than Memory	0
Enable TDC	N/A
Inforce Realtime	1
Event	N/A
etch Interleaved Pata	0
etch Meas Num amples	-1
etch Num Records	-1
etch Offset	0
etch Record umber	0
etch Relative To	Pretrigger
orizontal Min umber of Points	1000
nput Clock Source	No Source
nput Impedance	1000000
terchange Check	0
ine Number	N/A
ogical Name	
lanufacturer	National Instruments
laster Enable	0
1ax Input	10000000

<u>Max Realtime</u> Sampling Rate	10000000	
Max RIS Rate	N/A	
<u>Measurement Array</u> <u>Gain</u>	1	
<u>Measurement Array</u> <u>Offset</u>	0	
<u>Channel Based</u> High Ref	90	
<u>Channel Based Low</u> <u>Ref</u>	10	
<u>Channel Based Mid</u> <u>Ref</u>	50	
Filter Center Freq	1000000	
Filter Cutoff Freq	1000000	
Filter Order	2	
Filter Ripple	0.1	
Filter Taps	25	
<u>Percent Waveform</u> Transient	20	
Filter Type	Lowpass	
Filter Width	1000	
FIR Window	0	
Hysteresis Percent	2	
Interpolation Sampling Factor	2	
Last ACQ Histogram Size	256	
Other Channel	0	
Polynomial Interpolation Order	1	

<u>Time Histogram</u> <u>High Time</u>	0.0005
<u>Time Histogram</u> <u>High Volts</u>	10
<u>Time Histogram Low</u> Time	-0.0005
<u>Time Histogram Low</u> <u>Volts</u>	-10
<u>Time Histogram</u> <u>Size</u>	256
<u>Voltage Histogram</u> High Volts	10
<u>Voltage Histogram</u> Low Volts	-10
<u>Voltage Histogram</u> <u>Size</u>	256
Min Sample Rate	1000000
Model	NI PXI-5112, NI PCI-5112
Channel Count	2
Output Clock Source	No Source
Percentage Units Method	Base Top
PLL Locked	N/A
Points Done	N/A
Polarity	N/A
Probe Attenuation	1
Prog FIR Filter Taps	N/A
Record Arm Source	No Source
Records Done	N/A
Reference Clock Rate	N/A

<u>Reference Level</u> <u>Units</u>	Percentage	
Reference Position	50	
Resolution	8	
Resource Descriptor	device/session-specific	
RIS Method	Exact Num Avg	
RIS Num Avg	4	
<u>Sample Clock</u> Output Terminal	N/A	
<u>Sample Clock</u> Timebase Divisor	Real Time	
<u>Sample Clock</u> Timebase Rate	N/A	
<u>Sample Clock</u> Timebase Source	N/A	
Sample Mode	N/A	
Signal Format	N/A	
<u>Slave Trigger Delay</u>	0	
<u>SYNCOUT Clock</u> <u>Select</u>	N/A	
<u>Temperature</u>	N/A	
Test Mode Sin/Cos	N/A	
Time Per Record	0.001	
Trigger Coupling	DC	
Trigger Delay	0	
<u>Trigger From PFI</u> Delay	0	
Trigger From RTSI Delay	0	
Trigger From Star Delay	0	

Trigger Holdoff	0
Trigger Hysteresis	0.25
Trigger Impedance	100000
Trigger Level	0
<u>Trigger Output</u> <u>Event</u>	0
<u>Trigger Output</u> <u>Source</u>	No Source
Trigger Slope	Positive
Trigger Source	Immediate
Trigger To PFI Delay	0
<u>Trigger To RTSI</u> Delay	0
<u>Trigger To Star</u> Delay	0
Trigger Type	Immediate
<u>Trigger Window</u> <u>High Level</u>	0
<u>Trigger Window Low</u> <u>Level</u>	0
<u>Trigger Window</u> <u>Mode</u>	Entering
Vertical Coupling	DC
Vertical Offset	0
Vertical Range	10.39825

NI 5620/5621 Properties and Default Values

The following table lists the property default values for NI 5620/5621 digitizers. N/A indicates that the attribute is not supported by the device.

Property (Attribute)	Default Value
Acquisition Arm Source (Start Trigger Source)	Immediate
<u>Acquisition</u> <u>Start Time</u>	0
<u>Acquisition</u> <u>Type</u>	Normal
Actual Record Length	1000
<u>Actual Sample</u> <u>Rate</u>	100000
<u>Advance</u> Trigger Source	N/A
AGC Average Control	0
AGC Loop Gain <u>0 Exponent</u>	0
AGC Loop Gain <u>0 Mantissa</u>	0
AGC Loop Gain <u>1 Exponent</u>	0
AGC Loop Gain <u>1 Mantissa</u>	0
<u>AGC Lower</u> <u>Gain Limit</u>	6.0206
AGC_Threshold	845
<u>AGC Upper</u> <u>Gain Limit</u>	6.0206

<u>AOUT Parallel</u> <u>Output Source</u>	0
Arm Reference Trigger Source	N/A
Backlog	0
Binary Sample Width	16
BOUT Parallel Output Source	3
<u>Carrier NCO</u> <u>Center</u> Frequency	100000
<u>Carrier Phase</u> <u>Offset</u>	0
<u>Channel Count</u>	1
<u>Channel</u> Enabled	1
CIC Decimation	4
CIC Shift Gain	0
<u>Class Group</u> Capabilities	IviScopeBase, IviScopeWaveformMeasurement, IviScopeSampleMode, IviScopeAutoSetup, IviScopeTVTrigger
<u>Clock Sync</u> Pulse Source	No Source
<u>Combined</u> Decimation	0
<u>Coordinate</u> Converter Input	1
<u>Delay Before</u> Initiate	0
Description	National Instruments High Speed Digitizers Oscilloscope Instrument Driver
Discriminator	1

Discriminator	0	
Enable		
Discriminator FIR Decimation	1	
<u>Discriminator</u> FIR Input Source	0	
Discriminator FIR Symmetry	0	
Discriminator FIR Symmetry Type	0	
<u>Discriminator</u> Phase Multiplier	0	
Driver Prefix	niScope	
Driver Vendor	National Instruments	
Enable DC Restore	N/A	
Enable DDC	1	
Enable Dither	0	
Enable Records Greater Than Memory	0	
Enable TDC	N/A	
Enforce Realtime	1	
Event	N/A	
<u>Fetch</u> Interleaved Data	0	
Fetch Meas Num Samples	-1	

<u>Fetch Num</u> <u>Records</u>	-1	
Fetch Offset	0	
Fetch Record Number	0	
<u>Fetch Relative</u> To	Pretrigger	
Halfband Filter 1 Enable	1	
Halfband Filter 2 Enable	0	
Halfband Filter 3 Enable	0	
Halfband Filter 4 Enable	0	
<u>Halfband Filter</u> <u>5 Enable</u>	0	
<u>Halfband Filter</u> Bypass	1	
<u>Horizontal Min</u> <u>Number of</u> <u>Points</u>	1000	
Input Clock Source	No Source	
Input Impedance	50	
Interchange Check	0	
Line Number	N/A	
Logical Name	—	
<u>Manufacturer</u>	National Instruments	
Master Enable	0	

<u>Max Input</u> Frequency	10000000	
Max Realtime Sampling Rate	6400000	
Max RIS Rate	N/A	
Measurement Array Gain	1	
Measurement Array Offset	0	
<u>Channel Based</u> High Ref	90	
<u>Channel Based</u> Low Ref	10	
<u>Channel Based</u> <u>Mid Ref</u>	50	
Filter Center Frequency	1000000	
Filter Cutoff Frequency	1000000	
Filter Order	2	
Filter Ripple	0.1	
Filter Taps	25	
<u>Percent</u> <u>Waveform</u> <u>Transient</u>	20	
Filter Type	Lowpass	
Filter Width	1000	
FIR Filter Window	0	
Hysteresis Percent	2	
Interpolation Sampling	2	

Last ACQ	256	
Histogram Size		
Other Channel	0	
Polynomial Interpolation Order	1	
<u>Time Histogram</u> High Time	0.0005	
<u>Time Histogram</u> High Volts	10	
<u>Time Histogram</u> Low Time	-0.0005	
Time Histogram Low Volts	-10	
<u>Time Histogram</u> <u>Size</u>	256	
<u>Voltage</u> Histogram High Volts	10	
<u>Voltage</u> Histogram Low Volts	-10	
<u>Voltage</u> Histogram Size	256	
<u>Min Sample</u> Rate	1000000	
Model	NI PXI-5620, NI PXI-5621	
Channel Count	1	
<u>Output Clock</u> Source	No Source	
Percentage Units Method	Base Top	

PLL Locked	0	
Points Done	N/A	
<u>Polarity</u>	N/A	
Probe Attenuation	1	
Prog FIR Filter Decimation	1	
<u>Prog FIR Filter</u> <u>Taps</u>	1	
<u>Prog FIR_Filter</u> <u>Real/Complex</u>	0	
Prog FIR Symmetry	0	
Prog FIR Filter Symmetry Type	0	
<u>Q Input to</u> <u>Coordinate</u> <u>Converter</u>	0	
Record Arm Source	No Source	
Records Done	N/A	
Reference Clock Rate	N/A	
Reference Level Units	Percentage	
Reference Position	50	
Resampler Bypass	1	
<u>Resampler</u> Filter Mode	1	
<u>Resampler</u> NCO Divide	2	

<u>Resampler</u> <u>Output Pulse</u> <u>Delay</u>	16
Resampler Reference Divide	2
Resolution	12
<u>Resource</u> Descriptor	device/session-specific
RIS Method	Exact Num Avg
RIS Num Avg	4
<u>Sample Clock</u> Output Terminal	N/A
<u>Sample Clock</u> <u>Timebase</u> <u>Divisor</u>	Real Time
<u>Sample Clock</u> Timebase Rate	N/A
<u>Sample Clock</u> <u>Timebase</u> <u>Source</u>	N/A
Sample Mode	N/A
<u>Signal Format</u>	N/A
<u>Slave Trigger</u> <u>Delay</u>	0
Supported Instrument Models	NI PXI-5124, NI PCI-5124, NI PXI-5122, NI PCI-5122, NI PXI-5112, NI PCI-5112, NI PXI-5620, NI PXI-5621, NI PCI-5911, NI PXI-5102, NI PCI-5102, NI CPCI-5102, NI USB-5102, NI AT-5102, NI PCMCIA-5102
<u>SYNCOUT</u> Clock Select	0
Temperature	N/A
Test Mode	0

Time Per	0.001	
Record	0.001	
Timing NCO	-2147483648	
Center		
Frequency		
Timing NCO	0	
<u>Clear Phase</u>		
<u>Accum</u>		
Timing NCO	0	
Enable Offset		
Freq		
Timing NCO	0	
Freq Offset Bits		
Timing NCO	0A	
Phase Accum		
Load on Update		
Timing NCO	0	
Phase Offset		
Trigger	DC	
Coupling		
Trigger Delay	N/A	
Trigger From	0	
PFI Delay		
Trigger From	0	
RTSI Delay	-	
Trigger From	0	
<u>Star Delay</u>	-	
Trigger Holdoff	N/A	
Trigger	0.25	
Hysteresis		
Trigger	N/A	
Impedance		
Trigger Level	0	

<u>Trigger Output</u> <u>Event</u>	0
<u>Trigger Output</u> <u>Source</u>	No Source
Trigger Slope	Positive
Trigger Source	Immediate
<u>Trigger To PFI</u> Delay	0
<u>Trigger To RTSI</u> <u>Delay</u>	0
<u>Trigger To Star</u> <u>Delay</u>	0
Trigger Type	Immediate
<u>Trigger Window</u> <u>High Level</u>	0
<u>Trigger Window</u> Low Level	0
<u>Trigger Window</u> <u>Mode</u>	Entering
<u>Vertical</u> Coupling	AC (NI 5620), DC (NI 5621)
Vertical Offset	0
Vertical Range	2

NI 5911 Properties and Default Values

The following table lists the property default values for NI 5911 digitizers. N/A indicates that the attribute is not supported by the device.

Property (Attribute)	Default Value
Acquisition Arm Source	Immediate
(Start Trigger Source)	
Acquisition Start Time	0
Acquisition Type	Normal
Actual Record Length	1000
<u>Actual Sample Rate</u>	1000000
Advance Trigger Source	N/A
<u>Arm Reference Trigger</u> <u>Source</u>	N/A
Backlog	0
Binary Sample Width	8
Channel Count	1
Channel Enabled	0
<u>Class Group Capabilities</u>	IviScopeBase, IviScopeWaveformMeasurement, IviScopeSampleMode, IviScopeAutoSetup, IviScopeTVTrigger
Clock Sync Pulse Source	No Source
Combined Decimation	N/A
Delay Before Initiate	0
Description	National Instruments High Speed Digitizers Oscilloscope Instrument Driver
Driver Prefix	niScope
Driver Vendor	National Instruments
Enable DC Restore	N/A
Enable DDC	N/A
Enable Dither	N/A

Enable Records Greater Than Memory	N/A	
Enable TDC	N/A	
Enforce Realtime	1	
Event	N/A	
Fetch Interleaved Data	0	
Fetch Meas Num Samples	-1	
Fetch Num Records	-1	
Fetch Offset	0	
Fetch Record Number	0	
Fetch Relative To	Pretrigger	
Horizontal Min Number of Points	1000	
Input Clock Source	No Source	
Input Impedance	1000000	
Interchange Check	0	
Line Number	N/A	
Logical Name	—	
<u>Manufacturer</u>	National Instruments	
Master Enable	0	
Max Input Frequency	10000000	
Max Realtime Sampling Rate	10000000	
Max RIS Rate	N/A	
Measurement Array Gain	1	
Measurement Array Offset	0	
Channel Based High Ref	90	
Channel Based Low Ref	10	
Channel Based Mid Ref	50	

Filter Center Frequency	1000000	
Filter Cutoff Frequency	1000000	
Filter IIR Order	2	
Filter Ripple	0.1	
Filter Taps	25	
Percent Waveform Transient	20	
Filter Type	Lowpass	
Filter Width	1000	
FIR Filter Window	0	
Measurement Hysteresis Percent	2	
Measurement Interpolation Sampling Factor	2	
Measurement Last ACQ Histogram Size	256	
<u>Measurement Other</u> <u>Channel</u>	0	
Measurement_Polynomial Interpolation Order	1	
<u>Measurement Time</u> <u>Histogram High Time</u>	0.0005	
<u>Measurement Time</u> <u>Histogram High Volts</u>	10	
<u>Measurement Time</u> <u>Histogram Low Time</u>	-0.0005	
<u>Measurement Time</u> Histogram Low Volts	-10	
Measurement Time Histogram Size	256	
Measurement Voltage Histogram High Volts	10	

Measurement Voltage Histogram Low Volts	-10
Measurement Voltage Histogram Size	256
Min Sample Rate	1000000
Model	NI PCI-5911
Channel Count	1
Output Clock Source	No Source
Percentage Units Method	Base Top
PLL Locked	N/A
Points Done	1000
<u>Polarity</u>	N/A
Probe Attenuation	1
Record Arm Source	No Source
Records Done	1
Reference Clock Rate	N/A
Reference Level Units	Percentage
Reference Position	50
Resolution	8
Resource Descriptor	device/session-specific
RIS Method	Exact Num Avg
RIS Num Avg	4
Sample Clock Output Terminal	N/A
Sample Clock Timebase Divisor	Real Time
Sample Clock Timebase Rate	N/A
Sample Clock Timebase Source	N/A
Sample Mode	N/A

<u>Signal Format</u>	N/A	
Slave Trigger Delay	0	
SYNCOUT Clock Select	N/A	
Temperature	N/A	
Test Mode Sin/Cos	N/A	
Time Per Record	0.001	
Trigger Coupling	DC	
Trigger Delay	0	
Trigger From PFI Delay	0	
Trigger From RTSI Delay	0	
Trigger From Star Delay	0	
Trigger Holdoff	0	
Trigger Hysteresis	0.25	
Trigger Impedance	1000000	
Trigger Level	0	
Trigger Output Event	0	
Trigger Output Source	No Source	
Trigger Slope	Positive	
Trigger Source	Immediate	
Trigger To PFI Delay	0	
Trigger To RTSI Delay	0	
Trigger To Star Delay	0	
Trigger Type	Immediate	
<u>Trigger Window High</u> <u>Level</u>	0	
Trigger Window Low Level	0	
Trigger Window Mode	Entering	
Vertical Coupling	DC	
Vertical Offset	0	

Array Measurements

Histogram Measurements

Last Acq Histogram Multi Acq Time Histogram Multi Acq Voltage Histogram

Window Measurements

Blackman Window Hanning Window Flat Top Window Triangle_Window Hamming Window

Filter Measurements

Bessel IIR Filter FIR Windowed Filter Butterworth IIR Filter Chebyshev IIR Filter

Two Channel Measurements

Add Channels Multiply Channels Divide Channels Subtract Channels

Math Measurements

Derivative Multi_Acq_Average Array Integral Inverse Array Gain Array Offset

FFT Measurements

FFT Amp Spectrum (dB)

FFT Phase Spectrum

FFT Amp Spectrum (Volts RMS)

Resampling Measurements

Polynomial Interpolation

Last Acq Histogram

Notice that 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

Measurement Last Acq Histogram Size

Related Topics

Histogram Measurements

Multi Acq Time Histogram

Notice that 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 <u>Measurement Time Histogram Size</u> during the first time histogram measurement after the histogram is cleared.

Related Topics

Histogram Measurements

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 <u>Measurement Voltage Histogram Size</u> during the first voltage histogram measurement after the histogram is cleared.

Related Topics

Histogram Measurements

Blackman Window

A Blackman window is applied to the waveform using the following equation:

 $y[i] = waveform[i] \times (0.42659071 - 0.49656062cos(w) + 0.07684867cos(2w))$

where $w = (2_{\pi})i/n$ and *n* is the number of elements in the waveform.

Units

Volts

X Units Seconds

inputSize Related Topics

Window Measurements

Hanning Window

A Hanning window is applied to the waveform using the following equation:

 $y[i] = 0.5 \times waveform[i] \times [1 - \cos(w)]$

where $w = (2\pi)i/n$ and n = waveform size.

X Units Seconds

inputSize Related Topics

Window Measurements

Flat Top Window

A Flat Top window is applied to the waveform using the following equation:

 $y[i] = waveform[i] \times (0.2810639 - 0.5208972cos(w) + 0.1980399cos(2w))$ where $w = (2_{\pi})i/n$ and n is the waveform size.

volts

X Units Seconds

inputSize Related Topics

Window Measurements

Triangle Window

A triangular window is applied to the waveform using the following equation:

 $y[i] = waveform[i] \times (1 - |w|)$

where w = (2i - n)/n and *n* is the number of elements in the waveform.

X Units Seconds

inputSize Related Topics

Window Measurements

Hamming Window

A Hamming window is applied to the waveform using the following equation:

 $y[i] = x[i] [0.54 - 0.46\cos(w)]$

where $w = (2_{\pi})i/n$ and n = the number of elements in *x*.

X Units Seconds

inputSize Related Topics

Window Measurements

Bessel IIR Filter

The input waveform is filtered with a Bessel IIR algorithm. The type of the filter is specified by <u>Measurement Filter Type</u> as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The order of the filter is specified by <u>Filter IIR Order</u>. Also, a number of points equal to <u>Percent Waveform Transient</u> 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.

X Units

Seconds (nearly linear time delay)

inputSize × (1 – <u>Percent Waveform Transient</u> / 100) **Related Topics** <u>Filter Measurements</u> <u>Bessel Filters</u>

FIR Windowed 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 <u>Filter Taps</u> property. This number must be odd for highpass and bandstop filters. The window for the coefficients is set with the <u>FIR Window</u> property.

X Units

Seconds (linear time delay)

inputSize × (<u>Measurement Filter Taps</u> – 1)

Butterworth IIR Filter

The input waveform is filtered with a Butterworth IIR algorithm. The type of the filter is specified by <u>Filter Type</u> as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The order of the filter is specified by <u>Filter IIR Order</u>. Also, a number of points equal to <u>Percent Waveform Transient</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.

X Units

Seconds (nonlinear time delay)

inputSize × (1 – <u>Percent Waveform Transient</u> / 100) **Related Topics**

Filter Measurements

Butterworth Filters

Chebyshev IIR Filter

The input waveform is filtered with a Chebyshev IIR algorithm. The type of the filter is specified by <u>Filter Type</u> as any of the following values:

- lowpass
- highpass
- bandpass
- bandstop

The Chebyshev filter allows you to specify the passband ripple, <u>Filter</u> <u>Ripple</u>, in decibels. The order of the filter is specified by <u>Filter Order</u>. Also, a number of points equal to <u>Percent Waveform Transient</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.

X Units

Seconds (nonlinear time delay)

inputSize × (1 – <u>Percent Waveform Transient</u> / 100)

Related Topics

Filter Measurements

Chebyshev Filters

Add Channels

Adds the waveforms from two channels, where one channel is specified by the **channel** parameter to <u>niScope Fetch Measurement</u> (in the poly VI, select the Measurement Scalar DBL instance), and the other channel is specified by <u>Other Channel</u>. Any processing steps registered with <u>niScope Add Waveform Processing</u> are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

X Units Seconds

inputSize (on the channel specified by the measurement VI)

Multiply Channels

Multiplies the waveforms from two channels, where one channel is specified by the **channel** parameter to <u>niScope Fetch Measurement</u> (in the poly VI, select the Measurement Scalar DBL instance), and the other channel is <u>Measurement Other Channel</u>. Any processing steps registered with <u>niScope Add Waveform Processing</u> are completed for the "other channel" before this measurement is taken. The two channels used in this measurement must be different.

Volts²

X Units Seconds

inputSize (on the channel specified by the measurement VI)

Divide Channels

The channel specified by the **channel** parameter to <u>niScope Fetch</u> <u>Measurement</u> (in the poly VI, select the Measurement Scalar DBL instance is divided by the channel specified by <u>Measurement Other</u> <u>Channel</u>. Any processing steps registered with <u>niScope Add Waveform</u> <u>Processing</u> are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

None

X Units Seconds

inputSize (on the channel specified by the measurement VI)

Subtract Channels

The channel specified by <u>Other Channel</u> is subtracted from the specified by the **channel** parameter to <u>niScope Fetch Measurement</u> (in the poly VI, select the Measurement Scalar DBL instance). Any processing steps registered with <u>niScope Add Waveform Processing</u> are completed for the other channel before this measurement is taken. The two channels used in this measurement must be different.

Volts

X Units Seconds

inputSize (on the channel specified by the measurement VI)

Derivative

The differences in the waveform are computed using the formula:

 $y[i] = (waveform[i+1] - waveform[i-1]) / (2 \times dt)$, where dt is the time between two points.

Volts/second

X Units seconds

inputSize - 2

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 <u>niScope Clear Waveform</u> <u>Measurement Stats</u> with the **measurement function** parameter set to Multi Acq Average.

Volts

X Units Seconds

inputSize (during the first call to this measurement after clearing it)

Array Integral

The waveform is integrated using Simpson's rule, fitting a parabola to every 3 points with the equation:

 $y[2i + 1] = (waveform[2i] + 4 \times waveform[2i + 1] + waveform[2i + 2]) \times dt/3$

Volts × Seconds

X Units Seconds

inputSize – 2

Inverse

The inverse is computed using the following formula:

y[i] = 1.0/waveform[i]. If waveform[i] is zero, the inverse is zero.

1.0/volts

X Units Seconds

inputSize

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:

dB[*i*] = 20.0 × *log10* (*Amplitude*[*i*] / *Peak Amplitude*)

Decibels relative to peak

X Units

hertz

Next higher power of two than the inputSize divided by two.

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.

Radians

X Units

hertz

Resulting Array Size

Next higher power of two than inputSize divided by two.

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.

Volts RMS

X Units

hertz

Resulting Array Size

Next higher power of two than inputSize divided by two.

Polynomial Interpolation

Polynomial interpolation allows oversampling or undersampling a waveform using any order polynomial set by the property <u>Measurement Polynomial Interpolation Order</u>. As an example, an order of 1 corresponds to linear interpolation. The new number of points is determined by the input size times <u>Measurement Interpolation Sampling Factor</u>. 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.

Volts

X Units Seconds

Resulting Array Size

inputSize × <u>Measurement Interpolation Sampling Factor</u>

Scalar Measurements

RMS Voltage Measurements

AC Estimate Voltage Cycle RMS FFT Amplitude Voltage RMS

DC Voltage Measurements

Voltage Average Voltage Cycle Average DC Estimate

Area Measurements

<u>Area</u>

<u>Integral</u>

Cycle Area

Voltage Extrema Measurements

Voltage Min Voltage Low Voltage Max Voltage High Voltage Peak to Peak Voltage Amplitude Voltage Base Voltage Base Voltage Base to Top Overshoot Preshoot

Reference Levels

Low Ref Volts Mid Ref Volts High Ref Volts

Time Measurements

Fall TimePositive Duty CycleFalling Slew RateRise TimeNegative WidthRising Slew RatePositive WidthNegative Duty Cycle

Two Channel Measurements

<u>Phase Delay</u> <u>Time Delay</u>

Period and Frequency Measurements

Average Frequency FFT Frequency Average Period Frequency Period

Time Histogram Measurements

Time Hist HitsTime Hist Mean + 3 StdevTime Hist MaxTime Hist MedianTime Hist MeanTime Hist MeanTime Hist MinTime Hist Peak to PeakTime Hist ModeTime Hist Mean + StdevTime Hist New HitsTime Hist Mean + 2 StdevTime Hist Stdev

Voltage Histogram Measurements

Volt Hist Hits Volt Hist Median Volt Hist Max Volt Hist Min Volt Hist Mean Volt Hist Mean + Stdev Volt Hist Mean + Stdev Volt Hist Mean + 2 Stdev Volt Hist Peak to Peak Volt Hist Mean + 3 Stdev Volt Hist Stdev

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:

sqrt ([Σ processed waveform[i]²] / [numPoints × enbw × cg²]),

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.

Volts RMS

Related Topics

RMS Voltage Measurements

Voltage Cycle RMS

The number of points in a period is calculated using the equation:

pointsPerPeriod = int (period / dt + 0.5),

where *dt* is the time between two points and *int* is a VI that returns the integer portion of a floating-point number.

Volts RMS

Related Topics

RMS Voltage Measurements

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.

Volts RMS

Related Topics

RMS Voltage Measurements

Voltage RMS

Determined by the following equation: *Voltage RMS* = sqrt [(Σ *waveform*[*i*]²) / *numPoints*]

Volts RMS

Related Topics

RMS Voltage Measurements

Voltage Average

Determined by the following equation: Voltage Average = Σ waveform[i] / numPoints

Volts

Related Topics

DC Voltage Measurements

Voltage Cycle Average

The number of points in a period in volts is calculated using the equation: pointsPerPeriod = int (period / dt + 0.5),

where dt is the time between two points, and int is a VI that returns the integer portion of a floating-point number.

DC Estimate

A Hanning window is applied to give a "processed waveform" and the voltage average is calculated with the following equation:

voltage average = (Σ processed waveform[i]) / (cg × 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.

Volts

Related Topics

DC Voltage Measurements

Area

area = <u>voltage average</u> × *numPoints* × *delta time* between two points.

Volts × Seconds

Integral

Numerical integration is done using Simpson's rule.

Volts × seconds

Related Topics

Area Measurements

Cycle Area

Cycle area = voltage cycle average × pointsPerPeriod × dt where *pointsPerPeriod* is defined in the <u>voltage cycle average</u> description.

Volts × Seconds

Voltage Min

The waveform is searched for the minimum point in volts.

Related Topics

Waveform Measurement:Voltage Histogram:Low Volts

Short Name: V. Hist. Low Volts

The <u>last-acquisition histogram method</u> 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.

Volts

Related Topics

Voltage Max

The waveform is searched for its maximum point in volts.

Related Topics

Voltage High

The <u>last-acquisition histogram method</u> 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.

Volts

Related Topics

Voltage Peak to Peak

The <u>maximum voltage</u> minus the <u>minimum voltage</u> in volts.

Related Topics

Voltage Amplitude

The voltage high minus the voltage low in volts.

Related Topics

Voltage Base

If the histogram bin corresponding to <u>voltage low</u> has over five percent of the total hits, the voltage low result is returned. Otherwise, the <u>voltage</u> <u>minimum</u> 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).

Volts

Related Topics

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).

Volts

Voltage Base to Top

<u>Voltage top</u> minus <u>voltage base</u>.

Volts

Related Topics

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 × (*local maximum* - <u>voltage high</u>) / <u>voltage amplitude</u>.

If the first edge is negative sloped, *overshoot* = 100 × (<u>voltage low</u> - *local minimum*) / <u>voltage amplitude</u>.

Percentage

Related Topics

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 × (*local maximum* – <u>voltage high</u>) / <u>voltage amplitude</u>.

If the edge is positive sloped, *preshoot* = 100 × (voltage low – local minimum) / voltage amplitude.

Percentage

Related Topics

Low Ref Volts

The voltage corresponding to the low reference level. If the property <u>Reference Level Units</u> is set to Volts, the value of the <u>Channel Based</u> <u>Low Ref</u> property is returned. If the Reference Level Units property is set to Percentage, the voltage is calculated with the method specified by <u>Percentage Units Method</u>.

Volts Related Topics Related Topics <u>Reference Levels</u>

Mid Ref Volts

The voltage corresponding to the mid reference level. If the property <u>Reference Level Units</u> is set to Voltage, the value of the <u>Channel Based</u> <u>Mid Ref</u> property is returned. If the property Reference Level Units is set to Percentage, the voltage is calculated with the method specified by <u>Percentage Units Method</u>.

Volts

Related Topics

Reference Levels

High Ref Volts

The voltage corresponding to the high reference level. If the property <u>Reference Level Units</u> is set to Voltage, the value of the <u>Channel Based</u> <u>High Ref</u> property is returned. If the property is Percentage, the voltage is calculated with the method specified by <u>Percentage Units Method</u>.

Volts Related Topics Related Topics

Reference Levels

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 <u>Channel Based Low Ref</u> and <u>Channel Based High Ref</u>, and their default values are 10% and 90%.

Seconds

Related Topics

Time Measurements

Positive Duty Cycle

The <u>positive width</u> divided by the <u>period</u> times 100.

Percentage

Falling Slew Rate

The <u>low reference voltage</u> minus the <u>high reference voltage</u> is divided by the fall-time calculation. The result is always negative.

Volts/second

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 <u>Channel Based Low Ref</u> and <u>Channel Based High Ref</u>. Their default values are 10% and 90%.

Seconds

Related Topics

Time Measurements

Rising Slew Rate

The <u>high reference voltage</u> minus the <u>low reference voltage</u> is divided by the <u>rise-time</u> calculation.

Volts/second

Negative Width

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.

Seconds

Positive Width

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.

Seconds

Negative Duty Cycle

The negative width divided by the period times 100.

Percentage

Phase Delay

The <u>time delay</u> divided by the period (of the waveform on the channel specified by the measurement VI) times 360 degrees.

Degrees

Related Topics

Two Channel Measurements

Time Delay

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 <u>Measurement Other Channel</u> crosses its mid-reference level. The time delay is the time from rising edge to rising edge.

Note If you want to measure from falling edge to falling edge, you can invert the data from 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 one of the two channels.

The mid-reference level is stored on a per channel basis, and midreference levels do not need to be the same. All reference levels use a digital hysteresis.

Seconds

Related Topics

Two Channel Measurements

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.

Seconds

Frequency

1.0 divided by the period, in hertz.

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 <u>Channel Based Mid Ref</u>.

Seconds

Time Hist Hits

Number of points in the histogram.

Related Topics

Time Hist Mean + 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.

Percentage

Related Topics

Average Frequency

1.0 divided by the <u>average period</u>.

hertz

Time Hist Max

The highest bin value with at least one hit.

Seconds

Related Topics

Time Hist Median

The bin value where half the histogram hits are above it and half the histogram hits are below.

Seconds

Related Topics

Time Hist Mean

Histogram Mean = [Σ (bin hits × bin value)] / total hits. The bin value is the center time value of the histogram bin.

Seconds

Related Topics

Time Hist Min

The lowest bin value with at least one hit.

Seconds

Related Topics

Time Hist Peak to Peak

Histogram maximum minus the histogram minimum.

Seconds

Related Topics

Time Hist Mode

The bin value with the most hits. If there is a tie, the lower voltage or time value is returned.

Seconds

Related Topics

Time Hist Mean + 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.

Percentage

Related Topics

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 is a power of 2, so no zero padding occurs. This property can be fetched using the niScope Actual Record Length. Alternatively, coupling the digitizer for AC coupling solves the problem.

hertz

Related Topics

Period and Frequency Measurements

Time Hist New Hits

Number of points added to the histogram by the most recent acquisition. **Related Topics**

Time Histogram Measurements

Time Hist Mean + 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.

Percentage

Related Topics

Time Histogram Measurements

Time Hist Stdev

 $Histogram \ Stdev \ = \ \sqrt{\Sigma[bin \ hits \times (bin \ value - histogram \ mean) \land 2] / (total \ hits - 1)}$

Units

Seconds

Related Topics

Time Histogram Measurements

Volt Hist Hits

Number of points in the histogram.

Related Topics

Volt Hist Median

The bin value where half the histogram hits are above it and half the histogram hits are below.

Volts

Related Topics

Volt Hist Max

The highest bin value with at least one hit.

Volts

Related Topics

Voltage Histogram Measurements

Voltage Extrema Measurements

Volt Hist Min

The lowest bin value with at least one hit.

Volts

Related Topics

Volt Hist Mean

Histogram Mean = [((bin hits × bin value)] / total hits. The bin value is the center voltage value of the histogram bin.

Volts

Related Topics

Volt Hist Mode

The bin value with the most hits. If there is a tie, the lower voltage or time value is returned.

Volts

Related Topics

Volt Hist Mean + 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.

Percentage

Related Topics

Volt Hist New Hits

Number of points added to the histogram by the most recent acquisition. **Related Topics**

Volt Hist Mean + 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.

Percentage

Related Topics

Volt Hist Peak to Peak

Histogram maximum minus the histogram minimum.

Volts

Related Topics

Volt Hist Mean + 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.

Percentage

Related Topics

Volt Hist Stdev

 $Histogram \ Stdev = \sqrt{\Sigma[bin \ hits \times (bin \ value - histogram \ mean) \land 2]/(total \ hits - 1)}$

Units

Volts

Related Topics

Channel String Syntax

For most VIs, the input parameters **channel**, **channel name**, and **channelList** have the following options:

- a single channel, such as 0
- a list of channels, such as 0,1 or 3,2,1,0
- a range of channels, such as 0–7 or 0:7
- all channels, which is designated by the empty string

Exceptions for Fetch/Read VIs

If you use an empty string in a fetch or read VI to designate all channels, all of the channels on the device *must* be enabled. If all channels are not enabled, NI-SCOPE returns an error.



Note The order of channels in a list is important. The attributes are set on the channels in the order they occur in the list. For Fetch and Read VIs, the data for each channel is returned in the order the channels occur in the list.