



IVI™ Class Driver Help for LabVIEW™

January 2007 Edition, Part Number 370036E-01

This help file contains technical and programming support for the following National Instruments Interchangeable Virtual Instruments class drivers: IviDCPwr, IviDmm, IviFgen, IviPwrMeter, RFSigGen, IviScope, IviSpecAn, and IviSwtch. This help file also describes the fundamental and advanced terminology for these IVI class drivers. This help file is intended for use by developers with a working knowledge of LabVIEW.

Click on the IVI class-driver library you want to refer to:



[IviDCPwr
Power
Supply](#)



[IviDmm Digital Multimeter](#)



[IviFgen Function Generator](#)



[IviPwr
Meter](#)



[IviRFSigGen
RF Signal
Generator](#)



[IviSpecAn Spectrum
Analyzer](#)



[IviScope Oscilloscope](#)



[IviSw
tch](#)

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

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

[Conventions](#)—formatting and typographical conventions in this help file

[Related Documentation](#)

[Important Information](#)

[Technical Support and Professional Services](#)

To comment on National Instruments documentation, refer to the [National Instruments Web site](#).

© 2000–2007 National Instruments Corporation. All rights reserved.

Using Help

[Conventions](#)

[Navigating Help](#)

[Searching Help](#)

[Printing Help File Topics](#)

Conventions

This help file uses the following conventions:

- < > Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DBIO<3..0>.
- » The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
-  This icon denotes a note, which alerts you to important information.
- bold** Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names, emphasis, or an introduction to a key concept.
- green Underlined text in this color denotes a link to a help topic, help file, or Web address.
- italic* Italic text denotes variables or cross references. This font also denotes text that is a placeholder for a word or value that you must supply.
- monospace Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
- monospace bold** Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

Navigating Help (Windows Only)

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window or use the following toolbar buttons located above the tabs:

- **Hide**—Hides the navigation pane from view.
- **Locate**—Locates the currently displayed topic in the **Contents** tab, allowing you to view related topics.
- **Back**—Displays the previously viewed topic.
- **Forward**—Displays the topic you viewed before clicking the **Back** button.
- **Options**—Displays a list of commands and viewing options for the help file.

Printing Help File Topics (Windows Only)

Complete the following steps to print an entire book from the **Contents** tab:

1. Right-click the book.
2. Select **Print** from the shortcut menu to display the **Print Topics** dialog box.
3. Select the **Print the selected heading and all subtopics** option.
 **Note** Select **Print the selected topic** if you want to print the single topic you have selected in the **Contents** tab.
4. Click the **OK** button.

Printing PDF Documents

This help file may contain links to PDF documents. To print PDF documents, click the print button located on the Adobe Acrobat Viewer toolbar.

Searching Help (Windows Only)

Use the **Search** tab to the left of this window to locate content in this help file. If you want to search for words in a certain order, such as "related documentation," add quotation marks around the search words as shown in the example. Searching for terms on the **Search** tab allows you to quickly locate specific information and information in topics that are not included on the **Contents** tab.

Wildcards

You also can search using asterisk (*) or question mark (?) wildcards. Use the asterisk wildcard to return topics that contain a certain string. For example, a search for "prog*" lists topics that contain the words "program," "programmatically," "progress," and so on.

Use the question mark wildcard as a substitute for a single character in a search term. For example, "?ext" lists topics that contain the words "next," "text," and so on.



Note Wildcard searching will not work on Simplified Chinese, Traditional Chinese, Japanese, and Korean systems.

Nested Expressions

Use nested expressions to combine searches to further refine a search. You can use Boolean expressions and wildcards in a nested expression. For example, "example AND (program OR VI)" lists topics that contain "example program" or "example VI." You cannot nest expressions more than five levels.

Boolean Expressions

Click the  button to add Boolean expressions to a search. The following Boolean operators are available:

- **AND** (default)—Returns topics that contain both search terms. You do not need to specify this operator unless you are using nested expressions.
- **OR**—Returns topics that contain either the first or second term.
- **NOT**—Returns topics that contain the first term without the second term.
- **NEAR**—Returns topics that contain both terms within eight words of each other.

Search Options

Use the following checkboxes on the **Search** tab to customize a search:

- **Search previous results**—Narrows the results from a search that returned too many topics. You must remove the checkmark from this checkbox to search all topics.
- **Match similar words**—Broadens a search to return topics that contain words similar to the search terms. For example, a search for "program" lists topics that include the words "programs," "programming," and so on.
- **Search titles only**—Searches only in the titles of topics.

IVI Class Driver Help for LabVIEW Overview

This help file contains technical and programming support for the following National Instruments Interchangeable Virtual Instruments class drivers: IviDCPwr, IviDmm, IviFgen, IviPwrMeter, RFSigGen, IviScope, IviSpecAn, and IviSwtch. This book also describes the fundamental and advanced terminology for these IVI class drivers. The topics included in this book are intended for use by developers with a working knowledge of LabVIEW.

Click on the IVI class-driver library you want to refer to:



[IviDCPwr](#)
[Power](#)
[Supply](#)



[IviDmm](#)
[Digital](#)
[Multimeter](#)



[IviFgen](#)
[Function](#)
[Generator](#)



[IviPwr](#)
[Meter](#)



[IviRFSigGen](#)
[RF Signal](#)
[Generator](#)



[IviSpecAn](#)
[Spectrum](#)
[Analyzer](#)



[IviScope](#)
[Oscilloscope](#)



[IviSwtch](#)

Related Documentation

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

- [Application Notes](#)—A library with more than 100 short papers addressing specific topics such as creating and calling DLLs, developing your own instrument driver software, and porting applications between platforms and operating systems.
- [Measurement & Automation Explorer Help for IVI](#)—
This help file discusses how to configure your system with MAX.
- LabWindows/CVI Instrument Driver Developers Guide—This document describes guidelines for writing an IVI instrument driver. To download and use this document, click ni.com/manuals, and search for the LabWindows/CVI Instrument Driver Developers Guide.
- [IVI Foundation](#)—This web site provides information from the IVI Foundation, including the specifications for each instrument class.

Sample Programs

The IVI sample programs for LabVIEW are located in the LabVIEW\examples\instr\IviClass.llb directory. The sample programs are basic examples of class driver operations. Each example allows you to interactively configure the device. Some of the front panel default settings might not apply to all instruments. Refer to the VI and control help of the specific instrument driver for a description of valid values for your instrument.

IVI Inherent Capabilities

Expand this topic to view inherent VIs and Properties.

IVI Inherent Class Capabilities Overview

This section contains information and descriptions for inherent driver VIs that all class drivers support. Scroll down for information about class driver inherent VIs.

You can call any of these inherent VIs.

IVI Inherent VIs

Class/Panel Name	VI Name
Initialize/Close VIs	
Initialize	<u>ClassPrefix Initialize</u>
Initialize with Options	<u>ClassPrefix Initialize With Options</u>
Close	<u>ClassPrefix Close</u>
Property Access	
Invalidate All Attributes	<u>ClassPrefix Invalidate All Attributes</u>
Channel Access	
Get Channel Name	<u>ClassPrefix Get Channel Name</u>
Utility VIs	
Reset	<u>ClassPrefix Reset</u>
Reset With Defaults	<u>ClassPrefix Reset With Defaults</u>
Self-Test	<u>ClassPrefix Self-Test</u>
Disable	<u>ClassPrefix Disable</u>
Revision Query	<u>ClassPrefix Revision Query</u>
Error-Query	<u>ClassPrefix Error-Query</u>
Error Message	<u>ClassPrefix Error Message</u>
Coercion	
Get Next Coercion Record	<u>ClassPrefix Get Next Coercion Record</u>
Interchangeability Checking	
Get Next Interchange Warning	<u>ClassPrefix Get Next Interchange Warning</u>
Reset Interchange Check	<u>ClassPrefix Reset Interchange Check</u>
Clear Interchange Warnings	<u>ClassPrefix Clear Interchange Warnings</u>

Related Topics

[IVI Inherent Properties](#)

[IVI Status Codes](#)

ClassPrefix Clear Interchange Warnings

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Clear Interchange Warnings.

The class driver performs interchangeability checking if the [ClassPrefix Interchange Check](#) property is set to TRUE. This VI clears the list of current interchange warnings.



I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

Cluster **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

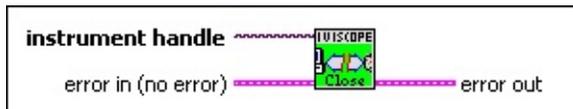
Cluster **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Close

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Close.



This VI performs the following operations:

- Closes the instrument I/O session.
- Destroys the instrument driver session and all of its properties.
- Deallocates any memory resources the driver uses.



Note After calling *ClassPrefix Close*, you cannot use the instrument driver again until you call [ClassPrefix Initialize](#).



instrument handle The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.



error in (no error) Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Disable

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Disable.



This VI places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.

 **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

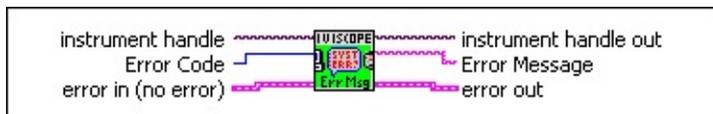
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Error Message

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Error Message.

This VI converts a status code returned by an instrument driver into a user-readable string.



I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

Error Code Pass the Status parameter that is returned from any of the instrument driver VIs.

Default Value: 0 (No error)

checkbox **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

checkbox **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

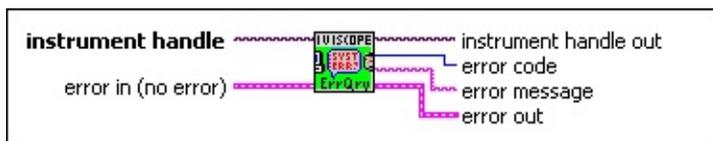
Error Message Returns a readable message string to you that

corresponds to the status code you specify.

ClassPrefix Error-Query

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Error-Query.



This VI queries the instrument and returns the instrument specific error information.

Generally, you call this VI after another VI in the instrument driver returns the IVI_ERROR_INSTRUMENT_STATUS (0xBFFA0001) error code. The driver returns this error code when the instrument's status register indicates that the instrument's error queue is not empty. The *ClassPrefix* Error Query VI extracts the first error out of the instrument's error queue. For instruments that have status registers but no error queue, the driver simulates an error queue in software.

I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

error code Returns the error code read from the instrument's error queue.

 **error message** Returns the error message string read from the instrument's error message queue.

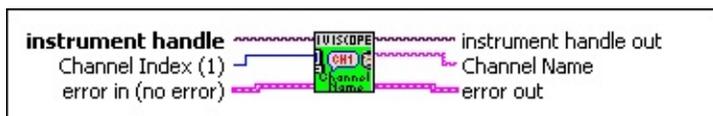
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Get Channel Name

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Get Channel Name.



This VI returns the specific driver defined channel name that corresponds to the one-based index that you specify. If the value you pass for the Index parameter is less than one or greater than the value of the Channel Count property, this VI returns an empty string in the Name parameter and returns an error.

- instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.
- index (1)** A one-based index that defines which name to return. The index must be less than or equal to the [ClassPrefix Channel Count](#) property.
- instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

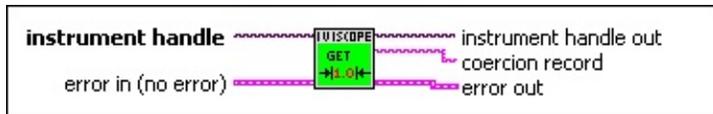
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
- output channel name** Returns the name of the Channel specified by the index parameter.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Get Next Coercion Record

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Get Next Coercion Record.



This VI obtains the coercion information associated with the IVI session. It retrieves and clears the oldest instance in which the specific driver coerced a value you specified to another value.

If you set the [ClassPrefix Record Value Coercions](#) property to TRUE, the specific driver keeps a list of all coercions it makes on ViInt32 or ViReal64 values you pass to instrument driver VIs. You use this VI to retrieve information from that list.

The VI returns an empty string in the **Coercion Record** parameter if no coercion records remain for the session.

 **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

coercion record Returns the next coercion record for the IVI session. If there are no coercion records, the VI returns an empty string.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Get Next Interchange Warning

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Get Next Interchange Warning.



This VI returns the interchangeability warnings associated with the IVI session. It retrieves and clears the oldest instance in which the class driver recorded an interchangeability warning. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. Use this VI to retrieve interchangeability warnings. Alternately, you may use NI Spy to view interchangeability warnings.

Interchangeability checking examines the properties in a capability group only if you specify a value for at least one property within that group. In general, the class driver generates interchangeability warnings when it encounters one of the following conditions:

- A property that affects the behavior of the instrument is in a state that you did not specify.
- You set a class-defined property to an instrument specific value.
- You set the value of a property that the class driver defines as read-only.
- The class driver encounters an error when it tries to apply a value to an extension property that your program never configures.

I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

interchange warning Returns the next interchange warning for the IVI session. If there are no interchange warnings, this parameter returns an empty string.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

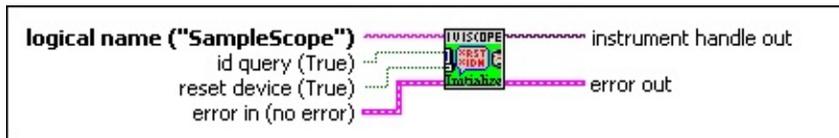
Related Topics

[Interchangeability Checking](#)

ClassPrefix Initialize

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Initialize With Options.



This VI performs the following initialization actions:

- Creates a new IVI instrument driver session.
- Opens a session to the specified device using the interface and address you specify for the **Resource Name** parameter.
- If the **ID Query** parameter is set to TRUE, this VI queries the instrument ID and checks that it is valid for this instrument driver.
- If the **Reset** parameter is set to TRUE, this VI resets the instrument to a known state.
- 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.



Note This VI creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads.



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.



logical name ("SampleFGen") Passes the logical name that

identifies the particular driver session to use. The driver session, in turn, identifies a specific driver and device and specifies the initial settings for the session. You configure the logical name, driver session, specific driver, and hardware asset with MAX.

If you want to use your program with a different physical instrument, you change the configuration of the logical name to use the driver session for the new physical instrument. You can change the initial settings for the session by changing the configuration of the driver session.

Default Values:

"SampleScope" for IviScope

"SampleDmm" for IviDmm

"SamplePower" for IviDCPwr

"SampleSwitch" for IviSwth

"SampleFgen" for IviFgen

- id query (true)** Specify whether you want the instrument driver to perform an ID Query.

Valid Range:

TRUE (1)Perform ID Query (Default Value)

FALSE (0)Skip ID Query

When you set this parameter to TRUE, the driver verifies that the instrument you initialize is a type that this driver supports.

Circumstances can arise where it is undesirable to send an ID Query command string to the instrument. When you set this parameter to FALSE, the VI initializes the instrument without performing an ID Query.

- reset device (true)** Specify whether you want the to reset the instrument during the initialization procedure.

Valid Range:

TRUE (1)Reset Device (Default Value)

FALSE (0)Don't Reset



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument driver that you use to identify the instrument in all subsequent instrument driver VI calls.



Note

1. The Initialize VI creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.
2. Avoid creating multiple concurrent instrument handles to the same physical instrument. Although you can create more than one instrument driver handle for the same resource, it is best not to do so.



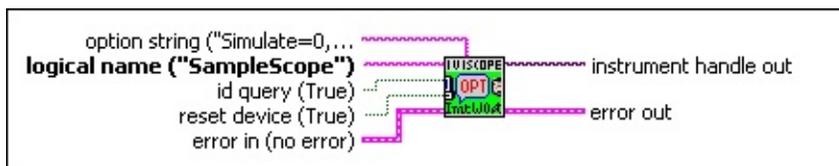
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Initialize With Options

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the *IviDmm* class driver, use the prefix *IviDmm* to create the correct VI name, *IviDmm Initialize With Options*. [Details](#).



- logical name ("SampleScope")** Passes the logical name that identifies the driver session to use. The driver session, in turn, identifies a specific instrument and specifies the initial settings for the session. You configure the logical session, specific driver, and hardware asset with MAX.

If you want to use your program with a different physical instrument, you can change the configuration of the logical name to use the driver session for the new instrument. You can change the initial settings for the session by changing the configuration of the driver session.

Default Values:

"SampleScope" for IviScope

"SampleDmm" for IviDmm

"SamplePower" for IviDCPwr

"SampleSwitch" for IviSwch

"SampleFgen" for IviFgen

- id query (True)** Specify whether you want the instrument driver to perform an ID Query.

Valid Range:

TRUE (1)—Perform ID Query (Default Value)

FALSE (0)—Skip ID Query

When you set this parameter to TRUE, the driver verifies that the instrument is present and ready for use.

initialize is a type that this driver supports.

Circumstances can arise where it is undesirable to send an ID Query to the instrument. When you set this parameter to FALSE, the VI initializes the instrument without performing an ID Query.

- reset device (True)** Specify whether you want the to reset the instrument initialization procedure.

Valid Range:

TRUE (1)—Reset Device (Default Value)

FALSE (0)—Do not Reset

-  **error in (no error)** The error in cluster can accept error information via the error in property previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- option string**
("Simulate=0,RangeCheck=0,QueryInstrStatus=0,Cache=1,InterchangeCheck=0")
You can use this parameter to set the initial value of certain properties. The following table lists the name and the properties you use in this parameter. Identify the property.

Name	Property
Range Check	<u>CLASSPREFIX_ATTR_RANGE_CHECK</u>
QueryInstrStatus	<u>CLASSPREFIX_ATTR_QUERY_INSTRUMENT_STATUS</u>
Cache	<u>CLASSPREFIX_ATTR_CACHE</u>
Simulate	<u>CLASSPREFIX_ATTR_SIMULATE</u>
RecordCoercions	<u>CLASSPREFIX_ATTR_RECORD_COERCIONS</u>
InterchangeCheck	<u>CLASSPREFIX_ATTR_INTERCHANGE_CHECK</u>

The format of this string is, "PropertyName=Value" where PropertyName is the name of the property and Value is the value to which the property will be set. To set multiple properties, separate their assignments with a comma.

If you pass NULL or an empty string for this parameter, the session uses the default values for the properties. You can override the default values by assigning a value explicitly in a string you pass for this parameter. You

do not have to specify all of the properties and may leave any of them out. If you do not specify one of the properties, its default value will be used.

The default values for the properties are shown below:

Name	Default Value
RangeCheck	TRUE
QueryInstrStatus	FALSE
Cache	TRUE
Simulate	FALSE
RecordCoercions	FALSE
InterchangeCheck	FALSE
DriverSetup	""

The following are valid values for ViBoolean properties:

True: 1, TRUE, or TRUE

False: 0, False, or FALSE

Default Value: ""

An example option string:

"Simulate=0,RangeCheck=1,QueryInstStatus=0,Cache=1,Spy=0,Intercha



Note Spying on IIVI API calls within NI-Spy is a feature that is only available if you have NI-Spy installed on your system. NI-Spy is installed with several NI software packages.

 **instrument handle out** The instrument driver that you use to identify the instrument in all subsequent instrument driver VI calls.



Notes

1. This VI creates a new session each time you invoke it, which is useful if you have multiple physical instances of the same type of instrument.
2. Avoid creating multiple concurrent instrument handles to the same physical instrument. Although you can create more than one instrument driver handle for the same resource, it is best not to do so.

 **error out** The error out cluster passes error or warning information

out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Initialize With Options Details

This VI performs the following initialization actions:

- Creates a new IVI instrument driver session.
- Opens a session to the specific driver using the logical name of the IVI virtual instrument.
- If the ID Query parameter is set to TRUE, this VI queries the instrument ID and checks that it is valid for this instrument driver.
- If the Reset parameter is set to TRUE, this VI resets the instrument to a known state.
- 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.



Note This VI creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads.



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.

ClassPrefix Invalidate All Attributes

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Invalidate All Attributes.



This VI invalidates the cached values of all properties for the session.

- instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Reset

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Reset.



This VI resets the instrument to a known state and sends initialization commands to the instrument. The initialization commands set instrument settings such as Headers Off, Short Command form, and Data Transfer Binary to the state necessary for the operation of the instrument driver.

Y/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

E/W **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

Y/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

E/W **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Reset Interchange Check

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Reset Interchange Check. [Details](#).



I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Reset Interchange Check Details

When developing a complex test system that consists of multiple test modules, it is generally a good idea to design the test modules so that

they can run in any order. To do so requires ensuring that each test module completely configures the state of each instrument it uses. If a particular test module does not completely configure the state of an instrument, the state of the instrument depends on the configuration from a previously executed test module. If you execute the test modules in a different order, the behavior of the instrument and therefore the entire test module is likely to change. This change in behavior is generally instrument specific driver and represents an interchangeability problem.

You can use this VI to test for such cases. After you call this VI, the interchangeability checking algorithms in the specific driver ignore all previous configuration operations. By calling this VI at the beginning of a test module, you can determine whether the test module has dependencies on the operation of previously executed test modules.

This VI does not clear the interchangeability warnings from the list of previously recorded interchangeability warnings. If you want to guarantee that the *ClassPrefix* Get Next Interchange Warning VI only returns those interchangeability warnings that are generated after calling this VI, you must clear the list of interchangeability warnings. You can clear the interchangeability warnings list by repeatedly calling the [ClassPrefix Get Next Interchange Warning](#) VI until no more interchangeability warnings are returned. If you are not interested in the content of those warnings, you can call the [ClassPrefix Clear Interchange Warnings](#) VI.

ClassPrefix Reset With Defaults

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Reset With Defaults.



This VI resets the instrument and applies initial user-specified settings from the Logical Name which was used to initialize the session.

 **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

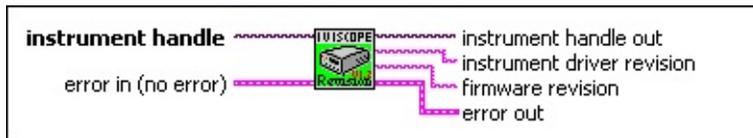
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Revision Query

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Revision Query.



Obtains the following information:

- The revision of the instrument driver.
- The firmware revision of the instrument you are currently using

I/O **instrument handle** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

instrument driver revision Returns the instrument driver software revision numbers in the form of a string.

firmware revision Returns the instrument firmware revision numbers in the form of a string.

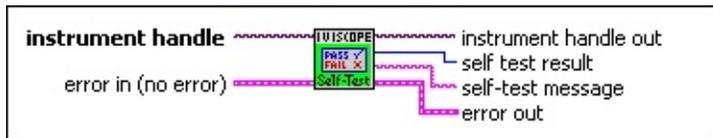
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

ClassPrefix Self-Test

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Self-Test.



This VI runs the instrument's self-test routine and returns the test result(s).

instrument handle The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

instrument handle out The instrument handle that you obtain from the [ClassPrefix Initialize](#) or [ClassPrefix Initialize With Options](#) VI. The handle identifies a particular instrument session.

self test result This control contains the value returned from the instrument self test. Zero means success. For any other code, refer to the device's operator manual.

Numeric result from self-test operation:

0 = no error (test passed)

non-zero = test failed

self-test message Returns the self-test response string from the

instrument. Refer to the device's documentation for an explanation of the string's contents.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

Inherent Properties



Note You access properties through the LabVIEW Property Node.

IVI Inherent Properties

Advanced Session Information

[Logical Name](#)

[Resource Descriptor](#)

[Driver Setup](#)

Class Driver Identification

[Description](#)

[Prefix](#)

[Vendor](#)

[Revision](#)

[Specification Major Version](#)

[Specification Minor Version](#)

Instrument Identification

[Manufacturer](#)

[Model](#)

[Firmware Revision](#)

Specific Driver Capabilities

[Supported Instrument Models](#)

[Group Capabilities](#)

[Function Capabilities](#)

[Channel Count](#)

Specific Driver Identification

[Description](#)

[Locator](#)

[Prefix](#)

[Revision](#)

[Major Version](#)

[Minor Version](#)

Vendor

User Options

Range Check

Query Instrument Status

Cache

Simulate

Record Value Coercions

Interchange Check

Spy

Use Specific Simulation

ClassPrefix Attribute Capabilities

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

This property is a comma-separated string that identifies the class properties that the specific instrument driver implements.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Capabilities»Attribute Capabilities**

ClassPrefix Cache

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether to cache the value of properties. When caching is enabled, the specific driver keeps track of the current instrument settings so that it can avoid sending redundant commands to the instrument. State caching can significantly increase execution speed. You specify a value for this property for the entire instrument driver. However, the specific driver can override this value for a particular property.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»User Options»Class Cache**

ClassPrefix Channel Count

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of available channels.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Channel Count

ClassPrefix Class Driver Description

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns a brief description of the class driver.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Class Driver Identification»Description**

ClassPrefix Class Driver Prefix

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the prefix for the class instrument driver. The prefix can be a maximum of 31 characters. The name of each user-callable VI in the class driver begins with this prefix. For example, if a class driver has a user-callable VI named IviDmm Initialize, then IviDmm is the prefix for that driver.

Property Node Path

ClassPrefix»Inherent IVI Settings»Class Driver Identification»Class Driver Prefix

ClassPrefix Class Driver Revision

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns a string that contains additional version information about the specific instrument driver. The returned value contains driver prefix and version. It also contains version information for compiler and components used in the driver build process.

Property Node Path

ClassPrefix»Inherent IVI Settings»Class Driver Identification»Revision

ClassPrefix Class Driver Vendor

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the name of the vendor that supplies the class driver.

Property Node Path

ClassPrefix»Inherent IVI Settings»Class Driver Identification»Class Driver Vendor

ClassPrefix Class Specification Major Version

Inherent Property

Data Type	Access	Applies-to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Specifies the major version number of the IVI class specification with which the class driver is compliant.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Class Driver Identification»Class Specification Major Version**

ClassPrefix Class Specification Minor Version

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the minor version number of the class instrument driver.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Class Driver Identification»Minor Version**

ClassPrefix Driver Setup

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	ClassPrefix Initialize With Options

Description

This property returns the Driver Setup string that you specified when initializing the driver.

Some cases exist where you might specify the instrument driver options at initialization time. An example of this is specifying a particular instrument model from among a family of instruments that the driver supports. This is useful when using simulation. You can specify driver-specific options through the DriverSetup keyword in the optionString parameter to the [ClassPrefix Initialize With Options](#) VI or through MAX.

If you do not specify a Driver Setup string, this property returns an empty string.

Property Node Path

ClassPrefix»Inherent IVI Settings»Driver Setup

ClassPrefix Firmware Revision

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	ClassPrefix Revision Query

Description

Returns an instrument specific string that contains the firmware revision information of the physical instrument. When simulating, the specific driver returns the value.

Property Node Path

ClassPrefix»Inherent IVI Settings»Instrument Identification»Firmware Revision

ClassPrefix Function Capabilities

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

This property is a comma-separated string that identifies the class functions that the specific instrument driver implements.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Capabilities»Function Capabilities**

ClassPrefix Group Capabilities

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

This property is a comma-separated string that identifies the instrument class and the class-extension groups that the specific instrument driver implements.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Capabilities»Group Capabilities**

ClassPrefix Interchange Check

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether the driver performs interchangeability checking. When this property is enabled, the class driver keeps a list of the interchangeability warnings it encounters. You view the interchangeability warnings with NI Spy. Alternatively, you can call [ClassPrefix Get Next Interchange Warning](#) VI to extract and delete the oldest record from the list.



Note If Class Prefix Interchange Check is set to TRUE and you are not using NI Spy, the interchange check queue is cleared by calling [ClassPrefix Get Next Interchange Warning](#) VI. If you do not clear the queue, the resulting list build-up appears as a memory leak.

The default value is FALSE. You can override this value by specifying a value with MAX. You can clear the interchangeability warning list by calling the [ClassPrefix Clear Interchange Warnings](#) VI.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Interchange Check

ClassPrefix Logical Name

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

You pass a logical name to the class driver initialization VI. MAX must contain an entry for the logical name. The logical name entry refers to a driver session in MAX. The driver session specifies a physical device and a specific instrument driver. By assigning the name of a different driver session to the logical name in MAX, you can exchange one instrument for another without modifying or recompiling your LabVIEW application.

This property indicates the logical name you specified when opening the current IVI session.

Property Node Path

ClassPrefix»Inherent IVI Settings»Advanced Session Information»Logical Name

ClassPrefix Manufacturer

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the name of the manufacturer of the instrument.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Instrument Identification»Manufacturer**

ClassPrefix Model

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the model number or name of the physical instrument.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Instrument Identification»Model**

ClassPrefix Query Instrument Status

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether the specific driver queries the instrument status after each VI call. Querying the instrument status is useful for debugging. After validating the program, you can set this property to FALSE to disable status checking and maximize performance. You specify a value for this property for the entire instrument driver. However, the driver can override this value for a particular property.

The default value is FALSE. You can override this value by specifying a value with MAX or by initializing the IVI session with the [ClassPrefix Initialize With Options](#) VI.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Query Instrument Status

ClassPrefix Range Check

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether to validate property values and VI parameters. Range checking parameters is very useful for debugging. After validating the program, you can set this property to FALSE to disable range checking and maximize performance.

The default value is TRUE. You can override this value by specifying a value with MAX or by initializing the IVI session with the [ClassPrefix Initialize With Options](#) VI.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Range Check

ClassPrefix Record Value Coercions

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether the IVI Engine keeps a list of the value coercions it makes for and properties. When this property is enabled, the class driver maintains a record of each coercion. You view the coercion records with NI Spy. Alternately, you call [ClassPrefix_Get Next Coercion Record](#) VI to extract and delete the oldest coercion record from the list.

The default value is FALSE. You can override this value by specifying a value with MAX or by initializing the IVI session with the [ClassPrefix Initialize With Options](#) VI.



Note If ClassPrefix Record Value Coercions is set to TRUE and you are not using NI Spy, the coercion queue is cleared by calling ClassPrefix Get Next Coercion Info. If you do not clear the queue, the resulting list build-up appears as a memory leak.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Record Value Coercions

ClassPrefix Resource Descriptor

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the resource descriptor that the driver uses to identify the physical device. If you initialize the driver with a logical name that is associated with a device, this property contains the resource descriptor that corresponds to the entry in MAX.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Advanced Session Information»Resource Descriptor**

ClassPrefix Simulate

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether to simulate instrument driver I/O operations. If simulation is enabled, specific driver VIs perform range checking, but they do not perform instrument I/O. The class driver returns simulated data for output VI parameters.

The default value is `FALSE`. You can override this value by specifying a value with the MAX or by initializing the IVI session with the [ClassPrefix Initialize With Options](#) VI.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Simulate

ClassPrefix Specific Driver Class Specification Major Version

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the major version number of the IVI class specification with which the specific driver is compliant.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Identification»Class Specification Major Version**

ClassPrefix Specific Driver Class Specification Minor Version

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the minor version number of the IVI class specification with which the specific driver is compliant.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Identification»Class Specification Minor Version**

ClassPrefix Specific Driver Description

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns a brief description of the specific driver.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Identification»Description**

ClassPrefix Specific Driver Locator

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the information that the class driver uses to find the specific driver module.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Identification»Specific Driver Locator**

ClassPrefix Specific Driver Prefix

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

The name of each user-callable function in the specific driver begins with this prefix. For example, if the Fluke 45 driver has a user-callable function named "FL45_init", then "FL45" is the prefix for that driver.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Identification»Specific Driver Prefix**

ClassPrefix Specific Driver Revision

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns a string that contains additional version information about the specific instrument driver. The returned value contains driver prefix and version. It also contains version information for compiler and components used in the driver build process.

Property Node Path

ClassPrefix»Inherent IVI Settings»Specific Driver Identification»Revision

ClassPrefix Specific Driver Vendor

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the name of the vendor that supplies the specific driver

Property Node Path

ClassPrefix»Inherent IVI Settings»Specific Driver Identification»Specific Driver Vendor

ClassPrefix Spy

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether the class driver uses NI Spy to record calls to class driver VIs.

The default value is TRUE. You can override this value by initializing the IVI session with the [ClassPrefix Initialize With Options VI](#).

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Spy

ClassPrefix Supported Instrument Models

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns a comma-separated list that identifies the instrument models with which the specific driver is compatible.

Property Node Path

***ClassPrefix*»Inherent IVI Settings»Specific Driver Capabilities»Supported Instrument Models**

ClassPrefix Use Specific Simulation

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	R/W	N/A	None	None

Description

Specifies whether the class driver or the specific driver generates the simulation data for output parameters. A value of FALSE specifies that the class driver generates the simulation data for output parameters. A value of TRUE specifies that the specific driver generates the simulation data for output parameters.

The default value is FALSE. You can override this value by specifying a value with MAX.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Use Specific Simulation

Interchangeability Checking

Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. Use the [ClassPrefix Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. Alternately, you may use NI Spy to view interchangeability warnings.

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the base capability group. Interchangeability checking is performed on extension capability groups when you have set a value for any property in the group. If you never set the value of any property of an extension capability group, interchangeability checking is not performed on that group.

In general, interchangeability warnings are generated if the following conditions are encountered:

- A property that affects the behavior of the instrument is not in a state that you specify. This can happen if your program does not configure the property or if your program configures the property but the value becomes invalid as a result of your program configuring a different property.
- You set or get an instrument specific property.
- You set an property for which the IVI class specification defines values to an instrument specific value.
- You pass an instrument specific value to a VI parameter for which the IVI class specification defines values.
- You attempt to set an property that the class defines as read-only. In some cases the class drivers define read-only properties that specific drivers might implement as read/write.
- The class driver encounters an error when it tries to disable an extension capability group that your program does not use. The purpose of disabling unused extension groups is to make your program behave the same regardless of whether the instruments you use implement the extension capability group. Some instruments that implement the extension capability group might not support the value to which the class driver attempts to set the property. In this case, the class driver queues an interchangeability warning instead of returning an error from the VI.

Related Topics

[ClassPrefix Clear Interchange Warnings](#)

[ClassPrefix Get Next Interchange Warning](#)

[ClassPrefix Reset Interchange Check](#)

Click the following links to learn more about interchangeability checking for a particular IVI class driver.

[IviDCPwrDC Power Supply](#)

[IviDmmDigital Multimeter](#)

[IviFgenFunction Generator](#)

[IviPwrMeterPower Meter](#)

[IviRFSigGenRF Signal Generator](#)

[IviScopeOscilloscope](#)

[IviSpecAnSpectrum Analyzer](#)

[IviSwtchSwitch](#)

IviDCPwr Interchangeability

The IviDCPwr class driver performs interchangeability checking when the [IviDCPwr Interchange Check](#) property is set to TRUE and you call one of the following VIs:

[IviDCPwr Initiate](#)

[IviDCPwr Measure](#)

In addition to the [general interchangeability checking rules](#), the IviDCPwr class defines additional rules and exceptions for each capability group.

IviDCPwrTrigger The driver needs to perform interchangeability checking on this group only when the VI performing interchangeability checking is IviDCPwr Initiate.

The IviDCPwr class does not define additional interchangeability rules or exceptions for the following extension capability groups: [IviDCPwrBase](#), [IviDCPwrSoftwareTrigger](#), and [IviDCPwrMeasurement](#).

IviDmm Interchangeability

The IviDmm class driver performs interchangeability checking when the [IviDmm Interchange Check](#) property is set to VI_TRUE and you call one of the following VIs:

[IviDmm_Initiate](#)

[IviDmm_Read](#)

[IviDmm_ReadMultiPoint](#)

In addition to the [general interchangeability checking rules](#), the IviDmm class defines additional rules and exceptions for each capability group.

IviDmmBase If the IviDmm Function property is set to IVIDMM_VAL_TEMPERATURE, the IviDmm Resolution property is not required to be in a user-specified state.

IviDmmAC Measurement If the IviDmm Function property is set to IVIDMM_VAL_AC_VOLTS, IVIDMM_VAL_AC_CURRENT, IVIDMM_VAL_AC_PLUS_DC_VOLTS, or IVIDMM_VAL_AC_PLUS_DC_CURRENT, then the following properties are required to be in a user-specified state:

- IviDmm AC Minimum Frequency
- IviDmm AC Maximum Frequency

IviDmm Frequency Measurement If the IviDmm Function property is set to IVIDMM_VAL_FREQ or IVIDMM_VAL_PERIOD, then the IviDmm Frequency Voltage Range property is required to be in a user-specified state.

IviDmm Temperature Measurement If the IviDmm Function property is set to IVIDMM_VAL_TEMPERATURE, the IviDmm Transducer Type property is required to be in a user-specified state.

IviDmm Thermocouple If the IviDmm Transducer Type property is set to IVIDMM_VAL_THERMOCOUPLE, the following properties are required to be in a user-specified state:

- IviDmm Thermocouple Type

- IviDmm Reference Junction Type
- IviDmm Fixed Reference Junction

**IviDmm
Resistance
Temperature
Device**

If the IviDmm Transducer Type property is set to IVIDMM_VAL_2_WIRE_RTD or IVIDMM_VAL_4_WIRE_RTD, then the following properties are required to be in a user-specified state:

- IviDmm RTD Alpha
- IviDmm RTD Resistance

**IviDmm
Thermistor**

If the IviDmm Transducer Type property is set to IVIDMM_VAL_THERMISTOR, the IviDmm Thermistor Resistance property is required to be in a user-specified state.

**IviDmm
MultiPoint**

If the IviDmm Sample Count property is set to 1, then IviDMM Sample Trigger property is not required to be in a user-specified state.

If the IviDmm Sample Count property is set to 1 or the IviDmm Sample Trigger property is set to a value other than IVIDMM_VAL_INTERVAL, then the IviDmm Sample Interval property is not required to be in a user-specified state.

The IviDmm class does not define additional interchangeability rules or exceptions for the following extension capability groups:

[IviDmmSoftwareTrigger](#), [IviDmmDeviceInfo](#), [IviDmmAutoRangeValue](#), [IviDmmAutoZero](#), and [IviDmmPowerLineFrequency](#).

IviFgen Interchangeability

The IviFgen class driver performs interchangeability checking when the [IviFgen Interchange Check](#) property is set to TRUE and you call the [IviFgen Initiate Generation VI](#).

In addition to the [general interchangeability checking rules](#), the IviFgen class defines additional rules and exceptions for each capability group.

IviFgenStdFunc If the IviFgen Output Mode property is not set to IVIFGEN_VAL_OUTPUT_FUNC, no properties in the IviFgenStdFunc extension group are required to be in a user-specified state. If the IviFgen Output Enabled property is set to FALSE on a channel, no properties in the IviFgenStdFunc extension group are required to be in a user-specified state on that channel.

If the IviFgen Waveform property is set to IVIFGEN_VAL_WFM_DC on a channel, the following properties are not required to be in a user-specified state on that channel:

- IviFgen Func Amplitude
- IviFgen Frequency
- IviFgen Start Phase

If the IviFgen Waveform property is set to IVIFGEN_VAL_WFM_SQUARE on a channel, the IviFgen Duty Cycle High property is required to be in a user-specified state on that channel.

IviFgenArb Waveform

If the IviFgen Output Mode property is not set to IVIFGEN_VAL_OUTPUT_ARB, no properties in the IviFgenArbWaveform extension group are required to be in a user-specified state.

If the IVIFGEN_ATTR_OUTPUT_ENABLED property is set to FALSE on a channel, no properties in the IviFgenArbWaveform extension group are required to be in a user-specified state on that channel.

IviFgenArb

If the IviFgen Output Mode property is not set to

Sequence IVIFGEN_VAL_OUTPUT_SEQ, no properties in the IviFgenArbSequence extension group are required to be in a user-specified state.

If the IviFgen Output Enabled property is set to FALSE on a channel, no properties in the IviFgenArbSequence extension group are required to be in a user-specified state on that channel.

IviFgenTrigger If the IviFgen Operation Mode property is not set to IVIFGEN_VAL_OPERATE_BURST, no properties in the IviFgenTrigger extension group are required to be in a user-specified state.

IviFgenInternal Trigger If the IviFgen Trigger Source property is not set to IVIFGEN_VAL_INTERNAL_TRIGGER, no properties in the IviFgenInternalTrigger extension group are required to be in a user-specified state.

IviFgenBurst If the IviFgen Operation Mode property is not set to IVIFGEN_VAL_OPERATE_BURST, no properties in the IviFgenBurst extension group are required to be in a user-specified state.

If the IviFgen Output Enabled property is set to FALSE on a channel, no properties in the IviFgenBurst extension group are required to be in a user-specified state on that channel.

IviFgenModulate AM If the IviFgen Output Enabled property is set to FALSE on a channel, no properties in the IviFgenModulateAM extension group are required to be in a user-specified state on that channel.

If the IviFgen AM Enabled property is not set to TRUE on a channel, the IviFgen AM Source property is not required to be in a user-specified state on that channel.

If the IviFgen AM Source is not set to IviFgen AM Internal on at least one channel, the following

properties are not required to be in a user-specified state.

- IviFgen AM Internal Depth
- IviFgen AM Internal Waveform
- IviFgen AM Internal Frequency

IviFgenModulate FM

If the IviFgen Output Enabled property is set to FALSE on a channel, no properties in the IviFgenModulateFM extension group are required to be in a user-specified state on that channel.

If the IviFgen FM Enabled property is not set to TRUE on a channel, the IviFgen FM Source property is not required to be in a user-specified state on that channel.

If the IviFgen FM Source is not set to IviFgen FM Internal on at least one channel, the following properties are not required to be in a user-specified state.

- IviFgen FM Internal Deviation
- IviFgen FM Internal Waveform
- IviFgen FM Internal Frequency

The IviFgen class does not define additional interchangeability rules or exceptions for the following extension capability groups: [IviFgenBase](#), [IviFgenSoftwareTrigger](#).

IviPwrMeter Interchangeability

The IviPwrMeter class driver performs interchangeability checking when the [IVIPWRMETER_ATTR_INTERCHANGE_CHECK](#) property is set to VI_TRUE and you call one of the following VIs:

- [IviPwrMeter_Initiate](#)
- [IviPwrMeter_Read](#)
- [IviPwrMeter_ReadChannel](#)

In addition to the [general interchangeability checking rules](#), the IviRFSigGen class defines additional rules and exceptions for the following capability groups.

IviPwrMeter ManualRange Supports power meters that can manually specify the upper and lower limits of the measurement range. The IviPwrMeterManualRange extension capability also includes VIs for configuring the measurement range.

IviPwrMeter InternalTrigger Supports power meters that can trigger internally on the measurement signal. It specifies to configure the internal trigger event source, trigger level, and the trigger slope. Also includes VIs for configuring these properties.

IviPwrMeter SoftwareTrigger Supports power meters that can initiate a measurement based on a software trigger signal. You can send a software trigger to cause the power meter to trigger a measurement.

IviPwrMeter AveragingCount Supports power meters that can filter a signal by averaging it a specified number of times in manual averaging mode. The IviPwrMeterAveragingCount group defines a property and VI to specify the averaging count.

The IviPwrMeter class does not define additional interchangeability rules or exceptions for the following extension capability groups:

[IviPwrMeterChannelAcquisition](#) [IviPwrMeterTriggerSource](#) [IviPwrMeterD](#)
[IviPwrMeterZeroCorrection](#) [IviPwrMeterCalibration](#) [IviPwrMeterR](#)

IviRFSigGen Interchangeability

The IviRFSigGen class driver performs interchangeability checking when the [IVIRFSIGGEN_ATTR_INTERCHANGE_CHECK](#) property is set to VI_TRUE and you call one of the following VIs: [IviRFSigGen_IsSettled](#) and [IviRFSigGen_WaitUntilSettled](#).

In addition to the [general interchangeability checking rules](#), the IviRFSigGen class defines additional rules and exceptions for each capability group.

IviRFSigGen ModulateAM

If the IVIRFSIGGEN_ATTR_OUTPUT_ENABLE property is set to False, properties in the IviRFSigGenModulateAM extension group need a user-specified state.

If the IVIRFSIGGEN_ATTR_AM_SOURCE prop not set to an external modulation source, the IVIRFSIGGEN_ATTR_AM_EXTERNAL_COUPL property does need not be in a user-specified st

If the IVIRFSIGGEN_ATTR_AM_ENABLED pro set to False, properties in the IviRFSigGenModu extension group need not be in a user-specified

IviRFSigGen ModulateFM

If the IVIRFSIGGEN_ATTR_OUTPUT_ENABLE property is set to False, properties in the IviRFSigGenModulateFM extension group need a user-specified state.

If the IVIRFSIGGEN_ATTR_FM_SOURCE prop not set to an external modulation source, the pr IVIRFSIGGEN_ATTR_FM_EXTERNAL_COUPL does need not be in a user-specified state.

If the IVIRFSIGGEN_ATTR_FM_ENABLED pro set to False, properties in the IviRFSigGenModu extension group need not be in a user-specified

IviRFSigGen ModulatePM

If the IVIRFSIGGEN_ATTR_OUTPUT_ENABLE property is set to False, properties in the IviRFSigGenModulatePM extension group need a user-specified state.

If the IVIRFSIGGEN_ATTR_FM_SOURCE prop not set to an external modulation source, the IVIRFSIGGEN_ATTR_PM_EXTERNAL_COUPL property does need not be in a user-specified state.

If the IVIRFSIGGEN_ATTR_PM_ENABLED prop set to False, properties in the IviRFSigGenModulatePM extension group need not be in a user-specified state.

IviRFSigGen ModulatePulse

If the IVIRFSIGGEN_ATTR_OUTPUT_ENABLE property is set to False, properties in the IviRFSigGenModulatePulse extension group need not be in a user-specified state.

If the IVIRFSIGGEN_ATTR_PULSE_MODULATION_SOURCE property is not set to External, all other properties in the IviRFSigGenModulatePulse extension group need not be in a user-specified state.

If the IVIRFSIGGEN_ATTR_FM_ENABLED prop set to False, properties in the IviRFSigGenModulatePulse extension group need not be in a user-specified state.

IviRFSigGen LF Generator

If the LFGenerator Output Enabled property is set to True OR

If LFGenerator Output Enabled property is set to True AND one of the following sources includes an LFGenerator source

- AM Source
- FM Source
- PM Source

AND any of the following are set to True

- AM Enabled
- FM Enabled
- PM Enabled

then the properties in the IviRFSigGenLFGene extension group shall be in a user-specified state

**IviRFSigGen
LFGeneratorOutput**

If the LFGenerator Output Enabled property is set to False, the LFGenerator Output Amplitude property shall not be in a user-specified state.

**IviRFSigGen
PulseGenerator**

If the Pulse Output Enabled property is set to True AND the Pulse Generator Source is set to Internal AND the Pulse Modulation Enabled property is set to True, the properties in the IviRFSigGenPulseGenerator extension group shall be in a user-specified state.

If the Pulse Trigger Source property is not set to Internal, the Pulse Internal Trigger Period property shall not be in a user-specified state.

If the Pulse Trigger Source property is not set to External, the Pulse External Trigger Delay property and the Pulse External Trigger Slope property shall not be in a user-specified state.

**IviRFSigGen
PulseDouble
Generator**

If the Pulse Double Enabled property is set to False, the properties in the IviRFSigGenPulseDoubleGenerator extension group shall not be in a user-specified state.

If the IviRFSigGenPulseGenerator extension group is not checked for interchangeability, the properties in the IviRFSigGenPulseDoubleGenerator extension group shall not be in a user-specified state.

extension group need not be in a user-specified

**IviRFSigGen
Pulse
Generator
Output**

If the Pulse Output Enabled property is set to False, all other properties in the IviRFSigGenPulseGeneratorOutput extension group need not be in a user-specified state.

**IviRFSigGen
Sweep**

If the Sweep Mode property is set to None, all other properties in the IviRFSigGenSweep extension group need not be in a user-specified state.

If the Output Enabled property is set to False, all other properties in the IviRFSigGenSweep extension group need not be in a user-specified state.

**IviRFSigGen
Frequency
Sweep**

If the Sweep Mode property is not set to Frequency Sweep, properties in the IviRFSigGenFrequencySweep extension group need not be in a user-specified

If the IviRFSigGenSweep extension group does to be checked for interchangeability, then properties in the IviRFSigGenFrequencySweep extension group need not be in a user-specified state.

**IviRFSigGen
Power
Sweep**

If the Sweep Mode property is not set to Power Sweep in the IviRFSigGenPowerSweep extension group, properties in the IviRFSigGenPowerSweep extension group need not be in a user-specified state.

If the IviRFSigGenSweep extension group does to be checked for interchangeability, then properties in the IviRFSigGenPowerSweep extension group need not be in a user-specified state.

**IviRFSigGen
Frequency
Step**

If the Sweep Mode property is not set to Frequency, properties in the IviRFSigGenFrequencyStep extension group need not be in a user-specified state.

If the IviRFSigGenSweep extension group does to be checked for interchangeability, then properties in the IviRFSigGenFrequencyStep extension group need not be in a user-specified state.

**IviRFSigGen
Power
Step**

If the Sweep Mode property is not set to Power, properties in the IviRFSigGenPowerStep extension group need not be in a user-specified state.

If the IviRFSigGenSweep extension group does to be checked for interchangeability, then properties in the IviRFSigGenPowerStep extension group need not be in a user-specified state.

**IviRFSigGen
List**

If the Sweep Mode property is not set to List, properties in the IviRFSigGenList extension group need not be in a user-specified state.

If the IviRFSigGenSweep extension group does to be checked for interchangeability, then properties in the IviRFSigGenList extension group need not be in a user-specified state.

**IviRFSigGen
ALC**

If the ALC Enabled property is set to False, properties in the IviRFSigGenALC extension group need not be in a user-specified state.

If the Output Enabled property is set to False, all properties in the IviRFSigGenALC extension group need not be in a user-specified state.

**IviRFSigGen
ReferenceOscillator**

If the Output Enabled property is set to False, all properties in the IviRFSigGenReferenceOscillator extension group need not be in a user-specified state.

**IviRFSigGen
SoftwareTrigger**

No additional interchangeability rules or exceptions are defined for the IviRFSigGenSoftwareTrigger cap extension group.

**IviRFSigGen
ModulateIQ**

If the Output Enabled property is set to False, all properties in the IviRFSigGenModulateIQ extension group need not be in a user-specified state.

If the IQ Enabled property is set to False, all properties in the IviRFSigGenModulateIQ extension group need not be in a user-specified state.

**IviRFSigGen
IQImpairment**

If the IQ Impairment Enabled property is set to False, all properties in the IviRFSigGenIQImpairment extension group need not be in a user-specified state.

If the IviRFSigGenModulateIQ extension group is needed to be checked for interchangeability, then all properties in the IviRFSigGenIQImpairment extension group need not be in a user-specified state.

**IviRFSigGen
ArbGenerator**

If the IQ Source property is not set to ArbGenerator, all properties in the IviRFSigGenArbGenerator extension group need not be in a user-specified state.

If the Arb Trigger Source property is not set to External, all properties in the Arb External Trigger Slope property extension group need not be in a user-specified state.

If the IviRFSigGenModulateIQ extension group is needed to be checked for interchangeability, then all properties in the IviRFSigGenArbGenerator extension group need not be in a user-specified state.

group need not be in a user-specified state.

IviRFSigGen

DigitalModulationBase

If the Output Enabled property is set to False, properties in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.

If the IQ Source property is not set to DigitalModulationBase, properties in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.

If the DigitalModulationBase Clock Source property is set to External, the External Clock Type property need not be in a user-specified state.

If the DigitalModulationBase Data Source property is set to PRBS, the DigitalModulationBase PRBS property need not be in a user-specified state.

If the DigitalModulationBase Data Source property is set to Bit Sequence, the DigitalModulationBase Bit Sequence property need not be in a user-specified state.

If the IviRFSigGenModulateIQ extension group is used, properties in the IviRFSigGenDigitalModulationBase extension group need to be checked for interchangeability, then properties in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.

IviRFSigGen

CDMABase

If the Output Enabled property is set to False, properties in the IviRFSigGenCDMABase extension group need not be in a user-specified state.

If the IQ Source attribute is not set to CDMABase, properties in the IviRFSigGenCDMABase extension group need not be in a user-specified state.

If the CDMA Clock Source is not set to External, the CDMA External Trigger Slope property need not be in a user-specified state.

If the IviRFSigGenModulateIQ extension group need to be checked for interchangeability, then properties in the IviRFSigGenCDMABase extension group need not be in a user-specified state.

IviRFSigGen TDMABase

If the Output Enabled property is set to False, properties in the IviRFSigGenTDMABase extension group need to be in a user-specified state.

If the IQ Source property is not set to TDMABase, properties in the IviRFSigGenTDMABase extension group need not be in a user-specified state.

If the TDMA Clock Source property is not set to the TDMA External Trigger Slope property need to be in a user-specified state.

If the IviRFSigGenModulateIQ extension group need to be checked for interchangeability, then properties in the IviRFSigGenTDMABase extension group need not be in a user-specified state.

The IviRFSigGen class does not define additional interchangeability rules or exceptions for the following extension capability groups:

[IviRFSigGenAnalogModulationSource](#) and [IviRFSigGenSoftwareTrigger](#).

IviScope Interchangeability

The IviScope class driver performs interchangeability checking when the [IviScope Interchange Check](#) property is set to TRUE and you call one of the following VIs:

[IviScope Initiate Acquisition](#)

[IviScope Read Waveform](#)

[IviScope Read Min Max Waveform](#)

[IviScope Read Waveform Measurement](#)

[IviScope Fetch Waveform Measurement](#)

In addition to the [general interchangeability checking rules](#), the IviScope class defines additional rules and exceptions for each capability group.

IviScopeBase	The IviScope Trigger Level property must be in a user-specified state only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_EDGE_TRIGGER or IVISCOPE_VAL_RUNT_TRIGGER.
IviScopeTVTrigger	<p>The driver performs interchangeability checking on the IviScopeTVTrigger group only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_TV_TRIGGER.</p> <p>The IviScope TV Trigger Line Number property must be in a user-specified state only if the application sets the IviScope TV Trigger Event property to IVISCOPE_VAL_TV_EVENT_LINE_NUMBER.</p>
IviScopeRuntTrigger	The driver performs interchangeability checking on the IviScopeRuntTrigger group only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_RUNT_TRIGGER.
IviScopeGlitchTrigger	The driver performs interchangeability checking on the IviScopeGlitchTrigger group

only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_GLITCH_TRIGGER.

IviScopeWidthTrigger The driver performs interchangeability checking on the IviScopeWidthTrigger group only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_WIDTH_TRIGGER.

IviScopeAcLineTrigger The driver performs interchangeability checking on the IviScopeAcLineTrigger group only if the application sets the IviScope Trigger Type property to IVISCOPE_VAL_AC_LINE_TRIGGER.

IviScopeWaveformMeas The IviScope Measurement Low Reference property must be in a state you specify is you request a waveform measurement that requires the low reference level, such as rise time, fall time, preshoot and overshoot.

The IviScope Measurement Mid Reference property must be in a user-specified state if you request a waveform measurement that requires the middle reference level, such as frequency, period, positive and negative pulse widths, and all cycle-based measurements.

The IviScope Measurement High Reference property must be in a user-specified state if you request a waveform measurement that requires the high reference level, such as rise time, fall time, preshoot and overshoot.

IviScopeMinMax Waveform The driver performs interchangeability checking on the IviScopeMinMaxWaveform group only if the application sets the IviScope Acquisition Type property to IVISCOPE_VAL_ENVELOPE or IVISCOPE_VAL_PEAK_DETECT.

The IviScope Number of Envelopes property

must be in a user-specified state only if the application sets the IviScope Acquisition Type property to IVISCOPE_VAL_ENVELOPE.

IviScopeContinuous Acquisition

Using this extension group is inherently non-interchangeable. This specification does not define the behavior of the instrument nor the data the instruments return while continuously acquiring the data.

IviScopeAverage Acquisition

The driver performs interchangeability checking on the IviScopeAverageAcquisition group only if the application sets the IviScope Acquisition Type property to IVISCOPE_VAL_AVERAGE.

IviScopeAutoSetup

Using this extension group is inherently non-interchangeable. The application behavior then depends on the internal instrument's algorithms for optimal acquisition and channel configuration.

The IviScope class does not define additional interchangeability rules or exceptions for the following extension capability groups:

[IviScopeProbeAutoSense](#), [IviScopeSampleMode](#), [IviScopeInterpolation](#), and [IviScopeTriggerModifier](#).

IviSpecAn Interchangeability

The IviSpecAn class driver performs interchangeability checking when the [IVISPECAN_ATTR_INTERCHANGE_CHECK](#) property is set to VI_TRUE and you call one of the following VIs: [IviSpecAn_Initiate](#) and [IviSpecAn_ReadyTrace](#).

In addition to the [general interchangeability checking rules](#), the IviSpecAn class defines additional rules and exceptions for the following capability groups:

IviSpecAn Marker

If the Marker Enabled property is set to False for a particular marker, then the Marker Position, Marker Trace, Marker Frequency Counter Enabled, Marker Frequency Counter Resolution, and Signal Track Enabled properties need not be in a user specified state for that marker.

If the Marker Enabled property is set to False for all markers, then the Marker Threshold and Peak Excursion properties need not be in a user specified state.

Marker Frequency Counter Resolution property need not be in a user specified state for that marker.

IviSpecAn ExternalTrigger

The properties of this extension group must be in a user specified state if and only if the Trigger Source property is set to External.

IviSpecAn VideoTrigger

The properties of this extension group must be in a user specified state if and only if the Trigger Source property is set to Video.

IviSpecAn ExternalMixing

If the External Mixer Enabled property is set to False, then all remaining group properties need not be in a user specified state.

If the External Mixer Bias Enabled property is set to False, then the External Mixer Bias and External Mixer Bias Limit properties need not be in a user specified state.

The IviSpecAn class does not define additional interchangeability rules or

exceptions for the following extension capability groups:

[IviSpecAnMultiTrace](#) [IviSpecAnTrigger](#) [IviSpecAnDisplay](#)

[IviSpecAnMarkerType](#) [IviSpecAnDeltaMarker](#) [IviSpecAnPreselector](#)

IviSwtch Interchangeability

The IviSwtch class driver performs interchangeability checking when the [IviSwtch Interchange Check](#) property is set to TRUE and you call one of the following VIs:

[IviSwtch Connect Channels](#)

[IviSwtch Set Path](#)

[IviSwtch Initiate Scan](#)

The IviSwtch class does not define any additional rules or exception beyond the the [general interchangeability checking rules](#).

Disabling Unused Extensions - Overview

You might develop test programs that do not use one or more of the extension groups that a class driver defines. Normally your program should work with instruments that do not implement the extension group as well as with those that do. However, if your program does not configure an extension group and the specific driver implements that extension group, the values of the attributes in the unused extension group are unknown. The attributes are likely to be set to the power-on settings of the device. The power-on settings often vary from instrument to instrument, so when you rely on a specific driver to implement an extension group, you might sacrifice interchangeability.

To accommodate instruments that implement extension groups that your program never configures, the class driver disables unused extensions by setting the extension groups to an interchangeable state. This interchangeable state for an extension group configures the extension group to have no effect on the behavior of the instrument. For example, the IviDmm base capabilities control DMMs that can take a single measurement. The IviDmm class defines a multipoint extension group that controls DMMs that can acquire multiple samples from multiple triggers. If you develop a program that uses only the IviDmm base capabilities with an instrument that implements the multipoint extension group, the IviDmm class driver sets the multipoint extension group attributes to an interchangeable state when you call the Initiate or Read functions for IviDmm.

To disable the multipoint extension group to the interchangeable state, the IviDmm class driver sets the trigger count attribute to 1 and the sample count attribute to 1. In this configuration, the multipoint extension group does not affect the instruments behavior. Therefore, you can run the program with instruments that implement only the IviDmm base capabilities as well as with instruments that implement the multipoint extension group.

If your program has ever set any of the values of an extension group, the class driver does not configure the extension group.

Disabling Unused Extensions

IviDCPwr Class Driver

The IviDCPwr driver does not disable unused extensions.

Disabling Unused Extensions

IviFgen Class Driver

When you call the [IviFgen_InitiateGeneration](#) VI, the class driver disables extension capability groups that the program does not use by setting properties of the unused extension group to values that disable that extension group.

- If you do not use the IviFgenModulateAM extension group for a channel, this driver sets the [IVIFGEN_ATTR_AM_ENABLED](#) property to VI_FALSE for that channel.
- If you do not use the IviFgenModulateFM extension group for a channel, this driver sets the [IVIFGEN_ATTR_FM_ENABLED](#) property to VI_FALSE for that channel.

Disabling Unused Extensions

IviDmm Class Driver

When you call the [IviDmm_Initiate](#), the [IviDmm_Read](#), or the [IviDmm_ReadMultiPoint](#) VIs, the class driver disables extension capability groups that the program does not use by setting properties of the unused extension group to values that disable that extension group.

The class driver disables extension capability groups as follows:

- If you do not use the IviDmmMultiPoint extension group, this driver sets the [IVIDMM_ATTR_SAMPLE_COUNT](#) and [IVIDMM_ATTR_TRIGGER_COUNT](#) properties to 1.
- If you do not use the IviDmmAutoZero extension group, this driver sets the [IVIDMM_ATTR_AUTO_ZERO](#) property to [IVIDMM_VAL_AUTO_ZERO_OFF](#).

Disabling Unused Extensions

IviPwrMeter Class Driver

When you call the [IviPwrMeter_init](#), the [IviPwrMeter_InitWithOptions](#), or the [IviPwrMeter_Reset](#) VI, the class driver disables extension capability groups that the program does not use by setting of the unused extension group to values that disable that extension group.

The class driver disables extension capability groups by setting the following to the noted default values:

Unused Extension Group	Property
IviPwrMeterChannelAcquisition	IVIPWRMETER_ATTR_CHANNEL_ENA
IviPwrMeterTriggerSource	IVIPWRMETER_ATTR_TRIGGER_SOU
IviPwrMeterDutyCycleCorrection	IVIPWRMETER_ATTR_DUTY_CYCLE
IviPwrMeterReferenceOscillator	IVIPWRMETER_ATTR_REFERENCE_O



Note

The following extension groups are not disabled by the IviPwrMeter driver:

- [IviPwrMeterInternalTrigger](#)
- [IviPwrMeterSoftwareTrigger](#)
- [IviPwrMeterAveragingCount](#)
- [IviPwrMeterZeroCorrection](#)
- [IviPwrMeterSoftwareTrigger](#)
- [IviPwrMeterManualRange](#)

Disabling Unused Extensions

IviRFSigGen Class Driver

When you call the [IviRFSigGen_init](#), the [IviRFSigGen_InitWithOptions](#), or the [IviRFSigGen_Reset](#) VI, the class driver disables extension capability groups that the program does not use by setting of the unused extension group to values that disable that extension group.

The class driver disables extension capability groups by setting the following to the noted default values:

Unused Extension Group	Property
IviRFSigGenModulateAM	IVIRFSIGGEN_ATTR_AM_ENABLE!
IviRFSigGenModulateFM	IVIRFSIGGEN_ATTR_FM_ENABLE!
IviRFSigGenModulatePM	IVIRFSIGGEN_ATTR_PM_ENABLE!
IviRFSigGenModulatePulse	IVIRFSIGGEN_ATTR_PULSE_MODU!
IviRFSigGenLFGeneratorOutput	IVIRFSIGGEN_ATTR_LF_GENERAT!
IviRFSigGenPulseDoubleGenerator	IVIRFSIGGEN_ATTR_PULSE_DOUB!
IviRFSigGenPulseOutputEnabled	IVIRFSIGGEN_ATTR_PULSE_OUTP!
IviRFSigGenSweep	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenFrequencySweep	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenPowerSweep	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenFrequencyStep	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenPowerStep	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenList	IVIRFSIGGEN_ATTR_SWEEP_MOD!
IviRFSigGenACL	IVIRFSIGGEN_ATTR_ALC_ENABLE!
IviRFSigGenModulateIQ	IVIRFSIGGEN_ATTR_IQ_ENABLED!
IviRFSigGenIQImpairment	IVIRFSIGGEN_ATTR_IQ_IMPAIRME!

Note

The following extension groups are not disabled by the IviRFSigGen driver:

- IviRFSigGenLFGenerator

- IviRFSigGenPulseGenerator
- IviRFSigGenReferenceOscillator
- IviRFSigGenSoftwareTrigger
- IviRFSigGenArbGenerator
- IviRFSigGenDigitalModulationBase
- IviRFSigGenCDMABase
- IviRFSigGenTDMABase

Disabling Unused Extensions

IviScope Class Driver

When you call the [IviScope_InitiateAcquisition](#), [IviScope_ReadWaveform](#), [IviScope_ReadMinMaxWaveform](#), or [IviScope_ReadWaveformMeasurement](#) VIs the class driver disables extension capability groups that the program does not use by setting properties of the unused extension group to values that disable that extension group.

The class driver disables extension capabilities as follows:

- If you do not use the IviScopeContinuousAcquisition extension group, this driver sets the [IVISCOPE_ATTR_INITIATE_CONTINUOUS](#) property to `VI_FALSE`.
- If you do not use the IviScopeInterpolation extension group, this driver sets the [IVISCOPE_ATTR_INTERPOLATION](#) property to the [IVISCOPE_VAL_NO_INTERPOLATION](#) value.
- If you do not use the IviScopeTriggerModifier extension group, this driver sets the [IVISCOPE_ATTR_TRIGGER_MODIFIER](#) property to the [IVISCOPE_VAL_NO_TRIGGER_MOD](#) value.

Disabling Unused Extensions

IviSpecAn Class Driver

When you call the [IviSpecAn_init](#), the [IviSpecAn_InitWithOptions](#), or the [IviSpecAn_Reset](#) VI, the class driver disables extension capability groups that the program does not use by setting properties of the unused extension group to values that disable that extension group.

The class driver disables extension capability groups by setting the following properties to the noted default values:

Unused Extension Group	Property
IviSpecAnMarker	IVISPECAN_ATTR_MARKER_ENABLED
IviSpecAnTrigger	IVISPECAN_ATTR_TRIGGER_SOURCE
IviSpecAnExternalMixer	IVISPECAN_ATTR_EXTERNAL_MIXER_ENABI



Note

The following extension groups are not disabled by the IviRFSigGen driver:

- [IviSpecAnMultiTrace](#)
- [IviSpecAnExternalTrigger](#)
- [IviSpecAn_VideoTrigger](#)
- [IviSpecAnDisplay](#)
- [IviSpecAnMarkerType](#)
- [IviSpecAnDeltaMarker](#)
- [IviSpecAnPreselector](#)

Disabling Unused Extensions

IviSwtch Class Driver

When you call the [IviSwtch_Connect_Channels](#), the [IviSwtch_SetPath](#), or the [IviSwtch_InitiateScan](#) function, the class driver disables extension capability groups that the program does not use by setting properties of the unused extension group to values that disable that extension group.

If you do not use the IviSwtchScanner extension group, the driver uses the following default values.

IVISWTCH_ATTR_SCAN_LIST	"" (Empty string)
IVISWTCH_ATTR_TRIGGER_INPUT	IVISWTCH_VAL_IMM
IVISWTCH_ATTR_SCAN_ADVANCED_OUTPUT	IVISWTCH_VAL_NON
IVISWTCH_ATTR_SCAN_DELAY	0

IVI Status Codes

IVI class driver functions can return error and warning values from several sets of status codes. Some status codes are unique to the IVI class driver functions. Other status codes are the same codes that VISA Library functions return. Still others are error or warning values that functions in specific instrument drivers return. Each set of status codes has its own numeric range.

Regardless of the source of the status code, 0 always indicates success, a positive value indicates a warning, and a negative value indicates an error.

Status Code Tables	Specific Error and Warning Codes for Class Drivers
Status Code Ranges	IviDmm
Default Values of Defined Constants	IviFgen
IVI Errors and Warnings	IviDCPwr
Common Instrument Driver Errors and Warnings	IviScope
Most-Often-Encountered VISA Errors and Warnings	IviSwth
	IviPwrMeter
	IviRFSigGen
	IviSpecAn

The following table defines the different ranges of status codes. The table lists the include files that contain the defined constants for the particular status codes.

Status Code Ranges

Status Code Type	Numeric Range (in Hex)	Include File
IVI Errors	BFFA0000 to BFFA1FFF	ivi.h
IVI Warnings	3FFA0000 to 3FFA1FFF	ivi.h
Class Driver Errors	BFFA2000 to BFFA3FFF	<i>IviClass.h</i>
Class Driver Warnings	3FFA2000 to 3FFA3FFF	<i>IviClass.h</i>
Specific Driver Errors	BFFA4000 to BFFA5FFF	<i>Prefix.h</i>
Specific Driver Warnings	3FFA4000 to 3FFA5FFF	<i>Prefix.h</i>
Common Instrument Driver Errors	BFFC0000 to BFFCFFFF	vpptype.h
Common Instrument Driver Warnings	3FFC0000 to 3FFCFFFF	vpptype.h
VISA Errors	BFFF0000 to BFFFFFFF	visa.h
VISA Warnings	3FFF0000 to 3FFFFFFF	visa.h

The Common Errors and Warnings are values that *VXIplug&play* defines and that specific instrument drivers return. They provide a consistent set of codes for error and warning conditions that are common among all instrument drivers. Each particular instrument driver defines its own set of Driver Errors and Warnings. The status codes values for one driver can overlap the status code values for other drivers.

The IVI class drivers include files define particular status codes as the unsigned sum of a base value and a decimal integer value. The following table lists the base values.

Default Values of Defined Constants

Status Code Type	Defined Constant for Base Value	Value
IVI Errors	IVI_ERROR_BASE	BFFA0000
IVI Warnings	IVI_WARN_BASE	3FFA0000
Class Driver Errors	IVI_CLASS_ERROR_BASE	BFFA2000
Class Driver Warnings	IVI_CLASS_WARN_BASE	3FFA2000
Specific Driver Errors	IVI_SPECIFIC_ERROR_BASE	BFFA4000
Specific Driver Warnings	IVI_SPECIFIC_WARN_BASE	3FFA4000

For example, if you pass an invalid attribute ID to an IVI class driver function, the function returns `IVI_ERROR_INVALID_ATTRIBUTE`, which `ivi.h` defines as `IVI_ERROR_BASE + 12`, or `0xBFFA000C`.

The following tables contain the IVI Status Codes, the Common Status Codes, and the most commonly used VISA Status Codes.

IVI Errors and Warnings

Status	Description
0	The call was successful.
BFFA0000	Unrecoverable failure.
BFFA0001	Instrument error detected. Call the ClassPrefix_error_query function and examine the error.
BFFA0002	File could not be opened.
BFFA0003	File is being read.
BFFA0004	File is being modified.
BFFA0005	Driver module file not found.
BFFA0006	Cannot open driver module file for reading.
BFFA0007	Driver module has invalid file format or contains invalid data.
BFFA0008	Driver module contains undefined references.
BFFA0009	Cannot find function in driver module.
BFFA000A	Failure loading driver module.
BFFA000B	The path name is invalid.
BFFA000C	Attribute ID not recognized.
BFFA000D	Attribute is read-only.
BFFA000E	Attribute is write-only.
BFFA000F	Invalid parameter.
BFFA0010	Invalid value for parameter or property.
BFFA0011	Function or method not supported.
BFFA0012	Attribute or property not supported.
BFFA0013	The enumeration value for the parameter is not supported.
BFFA0014	Invalid Type.
BFFA0015	The attribute and function parameter types do not match.
BFFA0016	The specified attribute already has a value waiting to be updated.
BFFA0017	The specified item already exists.
BFFA0018	Not a valid configuration.

BFFA0019	The requested item or value does not exist or is not available.
BFFA001A	The requested attribute value not known and cannot be determined.
BFFA001B	There is no range table for this attribute.
BFFA001C	The range table is invalid.
BFFA001D	A connection to the instrument has not been initialized.
BFFA001E	The class instrument driver has encountered underspecified instrument configurations that limit interchangeability.
BFFA001F	No channel table has been built for the session. The instrument driver must call Ivi_BuildChannelTable in its IviInit function.
BFFA0020	Channel or repeated capability name specified is not valid for the instrument.
BFFA0021	Unable to allocate system resource.
BFFA0022	Permission to access file was denied.
BFFA0023	Too many files opened.
BFFA0024	Unable to create temporary file in target directory.
BFFA0025	All temporary filenames already used in target directory.
BFFA0026	Disk is full.
BFFA0027	Configuration file was not found on disk.
BFFA0028	Cannot open configuration file.
BFFA0029	Error reading configuration file.
BFFA002A	Invalid <input type="checkbox"/> value in configuration file.
BFFA002B	Invalid <input type="checkbox"/> value in configuration file.
BFFA002C	Invalid <input type="checkbox"/> value in configuration file.
BFFA002D	Entry missing from configuration file.
BFFA002E	Initialization failed in driver DLL.
BFFA002F	Driver module could not be loaded because of an unresolved external reference.
BFFA0030	Cannot find CVI Run-Time Engine.

BFFA0031	Cannot open CVI Run-Time Engine.
BFFA0032	CVI Run-Time Engine has invalid format.
BFFA0033	CVI Run-Time Engine is missing one or more required functions.
BFFA0034	CVI Run-Time Engine initialization failed, probably because of insufficient memory.
BFFA0035	CVI Run-Time Engine could not be loaded because of an unresolved external reference.
BFFA0036	Failure loading CVI Run-Time Engine.
BFFA0037	Cannot open DLL to read exports.
BFFA0038	DLL file is corrupt.
BFFA0039	No export table in DLL.
BFFA003A	Unknown attribute name for initial setting in configuration file.
BFFA003B	Invalid attribute value for initial setting in configuration file.
BFFA003C	Memory pointer specified is not known.
BFFA003D	Unable to find any channel or repeated capability strings.
BFFA003E	The channel or repeated capability list contains two instances of the same name.
BFFA003F	The VirtualChannelNames item in the configuration file contains a duplicate virtual channel name.
BFFA0040	The VirtualChannelNames item in the configuration file contains an entry without a virtual channel name (nothing before the '=').
BFFA0041	The VirtualChannelNames item in the configuration file contains an invalid virtual channel name. Channel names must contain only alphanumerics, underscores, or an exclamation point.
BFFA0042	The VirtualChannelNames item in the configuration file contains a virtual channel name without an assigned channel string (i.e., nothing after '=').
BFFA0043	The VirtualChannelNames item in the configuration file contains a virtual channel name that is assigned to an unknown or invalid channel string.

BFFA0044	Channel or repeated capability name required.
BFFA0045	The channel or repeated capability name is not allowed.
BFFA0046	The attribute is not valid for the specified channel or repeated capability.
BFFA0047	This operation requires a channel– or repeated capability–based attribute. The specified attribute is not channel– or repeated capability–based.
BFFA0048	The channel has already been excluded for the specified attribute and cannot be re-included.
BFFA0049	The option string parameter contains an entry without a name.
BFFA004A	The option string parameter contains an entry without a value.
BFFA004B	The option string parameter contains an entry with an unknown option name.
BFFA004C	The option string parameter contains an entry with an unknown option value.
BFFA004D	This operation is valid only on a session created by a class driver.
BFFA004E	You cannot create a configuration file named 'ivi.ini'. That filename is reserved.
BFFA004F	There already is an entry of the same name in the run-time configuration.
BFFA0050	The index parameter is one-based. You must pass a number greater than or equal to 1.
BFFA0051	The index exceeds the number of items available.
BFFA0052	You cannot set the cache for an attribute that has the IVI_VAL_NEVER_CACHE flag.
BFFA0053	An instrument driver cannot export a ViAddr attribute to the end-user. Use the IVI_VAL_HIDDEN flag macro to make it a private attribute.
BFFA0054	Channel or repeated capability strings must contain only alphanumerics, underscores, or an exclamation point.

BFFA0055	The Prefix item in the configuration file does not match the specific driver's prefix.
BFFA0056	The necessary memory could not be allocated.
BFFA0057	Operation in progress.
BFFA0058	Null pointer passed for parameter or property.
BFFA0059	Unexpected response from the instrument.
BFFA005B	File not found.
BFFA005C	The file format is invalid.
BFFA005D	The instrument status is not available.
BFFA005E	Instrument ID Query failed.
BFFA005F	Instrument reset failed.
BFFA0060	Insufficient location information or resource not present in the system.
BFFA0061	The driver is already initialized.
BFFA0062	The simulation state cannot be changed.
BFFA0063	Invalid number of levels in selector.
BFFA0064	Invalid range in selector.
BFFA0065	Unknown name in selector.
BFFA0066	Badly-formed selector.
BFFA0067	Unknown physical selector.
BFFA1190	The session handle is not valid.
BFFA1198	The session handle is not valid.
BFFA11A0	Could not create thread local.
BFFA1200	The specified configuration store file could not be deserialized.
BFFA1201	A deserialize was attempted after a previous deserialize had already succeeded.
BFFA1202	The specified configuration store file could not be serialized.
BFFA1203	The session name or logical name could not be resolved to a session or driver session.

BFFA1204	The item does not exist in the global collection.
BFFA1205	An entry with name already exists in the collection.
BFFA1206	The registry entry for the master configuration store does not exist or the file could not be found.
BFFA1207	The item does not exist in the collection.
BFFA1208	The data component is not a valid data component.
BFFA1209	The element cannot be removed from the global collection when it is referenced in the local collections.
BFFA1232	The specified handle is invalid or of an incorrect type.
BFFA1233	The specified property ID is not valid for this function.
BFFA6000	Repeated Capability lists cannot be modified after attributes have been added to them.
BFFA6001	An attribute can only be restricted to a subset of a repeated capability once.
BFFA6002	The repeated capability table cannot be built because it already exists.
BFFA6003	The repeated capability has not been defined yet.
BFFA6004	The repeated capability name cannot be an empty or null string.
BFFA600D	The Config Server module is not present on the system.
3FFA0065	Identification query not supported.
3FFA0066	Reset operation not supported.
3FFA0067	Self test operation not supported.
3FFA0068	Error query operation not supported.
3FFA0069	Revision query not supported.

Common Instrument Driver Errors and Warnings

Status	Description
BFFC0001	Parameter 1 out of range, or error trying to set it.
BFFC0002	Parameter 2 out of range, or error trying to set it.
BFFC0003	Parameter 3 out of range, or error trying to set it.
BFFC0004	Parameter 4 out of range, or error trying to set it.
BFFC0005	Parameter 5 out of range, or error trying to set it.
BFFC0006	Parameter 6 out of range, or error trying to set it.
BFFC0007	Parameter 7 out of range, or error trying to set it.
BFFC0008	Parameter 8 out of range, or error trying to set it.
BFFC0011	Instrument failed the ID Query.
BFFC0012	Invalid response from instrument.
3FFC0101	Instrument does not have ID Query capability.
3FFC0102	Instrument does not have Reset capability.
3FFC0103	Instrument does not have Self-Test capability.
3FFC0104	Instrument does not have Error Query capability.
3FFC0105	Instrument does not have Revision Query capability.

Most-Often-Encountered VISA Errors and Warnings

Status	Description
BFFF0000	Miscellaneous or system error occurred.
BFFF000E	Invalid session handle.
BFFF0015	Timeout occurred before operation could complete.
BFFF0034	Violation of raw write protocol occurred.
BFFF0035	Violation of raw read protocol occurred.
BFFF0036	Device reported an output protocol error.
BFFF0037	Device reported an input protocol error.
BFFF0038	Bus error occurred during transfer.
BFFF003A	Invalid setup (properties are not consistent).
BFFF005F	No listeners condition was detected.
BFFF0060	This interface is not the controller in charge.
BFFF0067	Operation is not supported on this session.
3FFF0085	The status value you passed is unknown.

IviDCPwr Class Driver Reference for LabVIEW

Concepts

[IviDCPwr Terminology](#)

[IviDCPwr Class Driver Overview](#)

[IviDCPwr Extension Capability Groups](#)

[IviDCPwr Behavior Models](#)

IviDCPwr VI and Property Reference

[IviDCPwr VI Tree](#)

[IviDCPwr Properties by Group](#)

[IviDCPwr Property Value Definitions](#)

[IviDCPwr Error and Completion Codes](#)

Other Reference

[Interchangeability Checking](#)

[Simulation](#)

IviDCPwr Terminology

This document uses the following terms to describe the power supply's output: Voltage Level, OVP Limit, Current Limit, Current Limit Behavior, Constant Voltage Mode, Constant Current Mode, and Unregulated Mode.

Voltage Level—The DC voltage the power supply attempts to generate. You configure the voltage level with the [IviDCPwr Voltage Level](#) property.

OVP—OVP is an acronym for Overvoltage Protection.

OVP Limit and OVP Enabled—If the OVP limit is enabled, the power supply disables the output when the output voltage is greater than or equal to the OVP limit. You configure the OVP limit with the [IviDCPwr OVP Limit](#) property, and enable or disable the OVP limit with the [IviDCPwr OVP Enabled](#) property.

Current Limit and Current Limit Behavior—The current limit behavior determines the behavior of the instrument when the output current is greater than or equal to the current limit. When the current limit behavior is trip, the power supply disables the output when the output current is greater than or equal to the current limit. When the current limit behavior is regulate, the power supply restricts the output voltage such that the output current is not greater than the current limit. You configure the current limit and current limit behavior with the [IviDCPwr Current Limit](#) and [IviDCPwr Current Limit Behavior](#) properties.

Constant Voltage Mode—The power supply is said to be in the constant voltage mode when the power supply's output signal reaches the voltage level before it reaches the current limit. In the constant voltage mode, the power supply's output voltage remains constant at the voltage level and its output current can vary.

Constant Current Mode—The power supply is said to be in the constant current mode when the power supply's output signal reaches the current limit before it reaches the voltage level, and the current limit behavior is set to regulate. In the constant current mode, the power supply's output current remains constant at the current limit and its output voltage varies.

Unregulated Mode—The power supply is said to be in the unregulated mode when the power supply's output signal reaches neither the voltage level or the current limit. In the unregulated mode, the power supply's

output current and output voltage varies.

IviDCPwr Class Driver Overview

This instrument driver provides programming support for the IviDCPwr Class. The IviDCPwr class is designed to support typical DC power supplies as well as common extended functionality found in more complex instruments. The IviDCPwr class conceptualizes a DC power supply as an instrument capable of generating a DC power signal.

The IviDCPwr class driver divides instrument capabilities into [IviDCPwrBase capability group](#) and multiple extension capability groups. The IviDCPwrBase group supports DC power supplies capable of configuring the voltage level, current limit, current limit behavior, and over voltage protection limit, as well as enabling and disabling output channels. It also allows you to configure the output range in which the power supply operates, and to query the instrument to determine in what state it is operating.

Extension Capability Groups

The IviDCPwr class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Abbreviation	Description
IviDCPwrTrigger	[TRG]	For power supplies capable of making changes to the output signal based on a trigger event.
IviDCPwrSoftwareTrigger	[SWT]	For power supplies capable of making changes to the output signal based on a software trigger event.
IviDCPwrMeasurement	[MSR]	For power supplies capable of returning measurement characteristics of the output signal.

Related Topics

[IVI Inherent Class Capabilities](#)

IviDCPwr Extension Capability Groups

The IviDCPwr class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver driver that you use must implement those extensions.

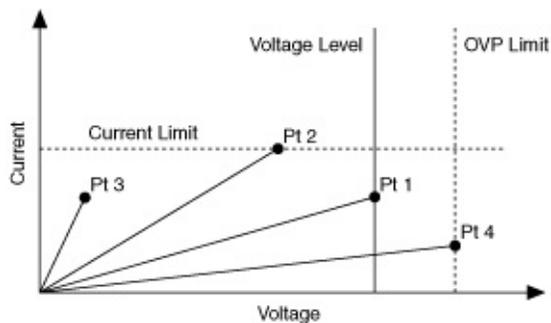
Group Name	Description
<u>IviDCPwrTrigger</u>	For power supplies capable of making changes to the output signal based on a trigger event.
<u>IviDCPwrSoftwareTrigger</u>	For power supplies capable of making changes to the output signal based on a software trigger event.
<u>IviDCPwrMeasurement</u>	For power supplies capable of returning measurement characteristics of the output signal.

IviDCPwrBase Capability Group

The IviDCPwrBase capability group supports the most basic DC power supply capabilities. You can enable or disable output channels, specify the DC voltage to generate, specify output limits, and control the behavior of the power supply when the output is greater than or equal to one of the limits.

This help file uses the following terms to describe the power supply's output: Voltage Level, OVP Limit, Current Limit, Current Limit Behavior, Constant Voltage Mode, Constant Current Mode, and Unregulated Mode. Refer to [IviDCPwr Terminology](#) for more information on these terms.

The signal that the power supply produces depends on the values of the voltage level, OVP limit, and current limit that you supply, and the impedance of the load to which the power supply is attached. Therefore, the power supply might not produce the exact voltage or current that you configure. The following diagram shows the possible output scenarios.



Power Supply Output Diagram

At Point 1, the power supply's output has reached the voltage level before it reached the current limit. This is an example of the power supply operating in the constant voltage mode. Note that any point on the vertical line defined by the voltage level would also cause the power supply to operate in the constant voltage mode.

At Point 2, the power supply's output has reached the current limit before it reached the voltage level. If the current limit behavior is set to regulate, this is an example of the power supply operating in the constant current mode. If the current limit behavior is set to trip, the power supply disables the output. Note that any point on the horizontal line defined by the current limit would also cause the power supply to operate in the constant current mode when the current limit behavior is set to regulate.

At point 3, the power supply's output has reached neither the voltage level or the current limit. This is an example of the power supply operating in the unregulated mode. Note that any point within the rectangle defined by the voltage level and current limit would also cause the power supply to operate in the unregulated mode.

At point 4, the power supply's output has reached the OVP limit. If OVP is enabled, the power supply disables the output. Note that any point on the vertical line defined by the OVP limit would also cause the power supply to disable the output when OVP is enabled.

IviDCPwrBase VIs

[IviDCPwr Configure OVP](#)

[IviDCPwr Configure Current Limit](#)

[IviDCPwr Configure Output Range](#)

[IviDCPwr Configure Output Enabled](#)

[IviDCPwr Query Output State](#)

[IviDCPwr Reset Output Protection](#)

[IviDCPwr Query Max Current Limit](#)

[IviDCPwr Query Max Voltage Level](#)

[IviDCPwr Voltage Level](#)

IviDCPwrBase Properties

[IviDCPwr OVP Enabled](#)

[IviDCPwr OVP Limit](#)

[IviDCPwr Current Limit Behavior](#)

[IviDCPwr Current Limit](#)

[IviDCPwr Output Enabled](#)

Related Topics

[IVI Inherent Functions](#)

[IVI Inherent Properties](#)

[IviDCPwrBase behavior model](#)

IviDCPwrMeasurement Extension Group

The IviDCPwrMeasurement extension group supports DC power supplies capable of returning output signal measurements such as voltage and current

IviDCPwrMeasurement VI

[IviDCPwr Measure](#)

IviDCPwrSoftwareTrigger Extension Group

The IviDCPwrSoftwareTrigger extension group supports DC power supplies capable of changing the output signal based on a software trigger event.

IviDCPwrSoftwareTrigger VI

[IviDCPwr Send Software Trigger](#)

IviDCPwrTrigger Extension Group

The IviDCPwrTrigger extension group supports DC power supplies capable of changing the output signal based on a trigger event.

Refer to the [IviDCPwrTrigger behavior model](#).

IviDCPwrTrigger VIs

[IviDCPwr Configure Trigger Source](#) [TRG]

[IviDCPwr Configure Triggered Voltage Level](#) [TRG]

[IviDCPwr Configure Triggered Current Limit](#) [TRG]

[IviDCPwr Initiate](#) [TRG]

[IviDCPwr Abort](#) [TRG]

IviDCPwrTrigger Properties

IviDCPwr Trigger Source

IviDCPwr Triggered Current Limit

IviDCPwr Triggered Voltage Level

IviDCPwr Behavior Models

[IviDCPwrBase](#)

[IviDCPwrMeasurement](#)

[IviDCPwrSoftwareTrigger](#)

[IviDCPwrTrigger](#)

IviDCPwrBase Behavior Model

After you call the [IviDCPwr Initialize](#), [IviDCPwr Initialize With Options](#), or [IviDCPwr Reset](#) VIs, the power supply produces a power signal based on its current configuration.

All changes to the power supply's IviDCPwrBase VIs and properties take place immediately.

IviDCPwrMeasurement Behavior Model

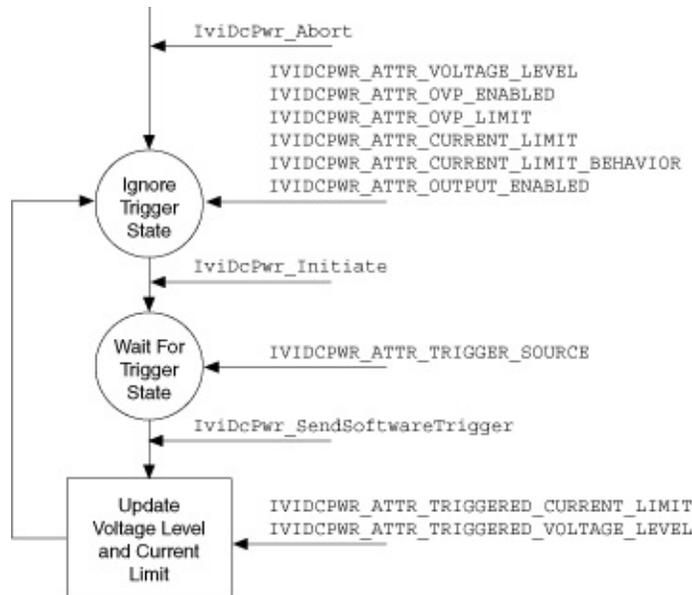
The IviDCPwrMeasurement Extension Group follows the behavior model of the [IviDCPwrBase](#) capability group. The only modification to the behavior model from the IviDCPwrBase capability group is the ability to take measurement on the output signal.

IviDCPwrSoftwareTrigger Behavior Model

The IviDCPwrSoftwareTrigger Extension Group follows the behavior model of the [IviDCPwrTrigger](#) capability group. The only modification to the behavior model from the IviDCPwrTrigger capability group is the ability to send software triggers.

IviDCPwrTrigger Behavior Model

The following behavior model shows the relationship between the IviDCPwrTrigger capability group and power supply behavior.



After you call the [IviDCPwr Initialize](#), [IviDCPwr Initialize With Options](#), or [IviDCPwr Reset](#) VIs., the power supply enters the *ignore trigger* state.

In the *ignore trigger* state, the properties of the [IviDCPwrBase capability group](#) determine the signal that the power supply produces.

IviDCPwrTrigger properties can be set, but do not affect the output signal.

Calling [IviDCPwr Initiate](#) VI moves the power supply to the wait for trigger state.

In the *wait for trigger* state, the power supply waits for a trigger on the trigger source specified by the [IviDCPwr Trigger Source \[TRG\]](#) property. After the power supply receives a trigger, it sets the voltage level to the value of the [IviDCPwr Triggered Voltage Level \[TRG\]](#) property, and the current limit to the value of the [IviDCPwr Triggered Current Limit \[TRG\]](#) property. It then returns to the *ignore trigger* state.

After the changes in output occur, the [IviDCPwr Triggered Voltage Level \[TRG\]](#) and [IviDCPwr Current Limit](#) properties reflect the power supply's new configuration.

Calling [IviDCPwr Abort](#) VI moves the power supply from its current state to the *ignore trigger* state. If the power supply has not yet responded to a trigger, no change occurs to the voltage level or current limit.

IviDCPwr VI Tree

The VI tree for the IviDCPwr class driver, including IVI and VXI*plug&play* required VIs, is shown below.



Note You use the LabVIEW Property Node to get and set properties.

IviDCPwr VI Tree

Name or Class	VI Name	Required By
Initialize	<u>IviDCPwr Initialize</u>	IVI
Initialize with Options	<u>IviDCPwr Initialize With Options</u>	IVI
Configuration VIs		
Configure Voltage Level	<u>IviDCPwr Configure Voltage Level</u>	IviDCPwrBase
Configure OVP	<u>IviDCPwr Configure OVP</u>	IviDCPwrBase
Configure Current Limit	<u>IviDCPwr Configure Current Limit</u>	IviDCPwrBase
Configure Output Range	<u>IviDCPwr Configure Output Range</u>	IviDCPwrBase
Configure Output Enabled	<u>IviDCPwr Configure Output Enabled</u>	IviDCPwrBase
Triggering		
Configure Trigger Source [TRG]	<u>IviDCPwr Configure Trigger Source</u>	IviDCPwrTrigger
Configure Triggered Voltage Level [TRG]	<u>IviDCPwr Configure Triggered Voltage Level</u>	IviDCPwrTrigger
Configure Triggered Current Limit [TRG]	<u>IviDCPwr Configure Triggered Current Limit</u>	IviDCPwrTrigger
Action		
Initiate [TRG]	<u>IviDCPwr Initiate</u>	IviDCPwrTrigger
Abort [TRG]	<u>IviDCPwr Abort</u>	IviDCPwrTrigger

Send SoftwareTrigger	<u>IviDCPwr Send Software Trigger</u>	IviDCPwrSoftwareTrigger
Query Max Current Limit	<u>IviDCPwr Query Max Current Limit</u>	IviDCPwrBase
Query Max Voltage Level	<u>IviDCPwr Query Max Voltage Level</u>	IviDCPwrBase
Query Output State	<u>IviDCPwr Query Output State</u>	IviDCPwrBase
Reset Output Protection	<u>IviDCPwr Reset Output Protection</u>	IviDCPwrBase
Measure	<u>IviDCPwr Measure</u>	IviDCPwrMeasurement
Utility		
Reset	<u>IviDCPwr Reset</u>	IVI
Reset With Defaults	<u>IviDCPwr Reset With Defaults</u>	VPP
Self-Test	<u>IviDCPwr Self-Test</u>	IVI
Disable	<u>IviDCPwr Disable</u>	VPP
Invalidate All Attributes	<u>IviDCPwr Invalidate All Attributes</u>	VPP
Revision Query	<u>IviDCPwr Revision Query</u>	IVI
Error-Query	<u>IviDCPwr Error-Query</u>	IVI
Error Message	<u>IviDCPwr Error Message</u>	IVI
Get Channel Name	<u>IviDCPwr Get Channel Name</u>	VPP
Interchangeability Info		

Get Next Interchange Warning	IviDCPwr Get Next Interchange Warning	IVI
Reset Interchange Check	IviDCPwr Reset Interchange Check	IVI
Clear Interchange Warnings	IviDCPwr Clear Interchange Warnings	IVI
Coercion Info		
Get Next Coercion Record	IviDCPwr Get Next Coercion Record	IVI
Close	IviDCPwr Close	IVI

Related Topics

[IviDCPwr Properties](#)

[IVI Inherent VIs](#)

IviDCPwr Properties



Note You use the LabVIEW Property Node to get and set properties.

Base Properties

[IviDCPwr Voltage Level](#)

[IviDCPwr OVP Enabled](#)

[IviDCPwr OVP Limit](#)

[IviDCPwr Current Limit Behavior](#)

[IviDCPwr Current Limit](#)

[IviDCPwr Output Enabled](#)

IviDCPwrTrigger Properties

IviDCPwr Trigger Source

IviDCPwr Triggered Current Limit

IviDCPwr Triggered Voltage Level

IviDCPwr Property Value Definitions

The following table defines values for all IviDCPwr class property constants.



Note In the following table, the literal string `IVIDCPWR_ATTR_` precedes the property name and the literal string `IVIDCPWR_VAL` precedes the value extension name.

IviDCPwr Property Defined Values

Property	Defined Value	Defined Value	Defined Value
CURRENT LIMIT BEHAVIOR	Regulate	0	The power supply regulates the output current at the value of the IviDCPwr Current Limit property when the output reaches or exceeds that value.
	Trip	1	The power supply disables the output when the output current reaches or exceeds the value of the IviDCPwr Current Limit property.
TRIGGER SOURCE	Immediate	0	The power supply does not wait for a trigger before changing the output signal.
	External	1	The power supply waits for an external trigger before changing the output signal.
	Software	2	The power supply waits for the IviDCPwr Send Software Trigger VI to execute before changing the output signal.
	TTL0	3	The power supply waits for a trigger on the TTL 0 line before changing the output signal.
	TTL1	4	The power supply waits for a trigger on the TTL 1 line before changing the output signal.
	TTL2	5	The power supply waits for a trigger on the TTL 2 line before changing the output signal.
	TTL3	6	The power supply waits for a trigger on the TTL 3 line before changing the output signal.
	TTL4	7	The power supply waits for a trigger on the TTL 4 line before changing the

			output signal.
	TTL5	8	The power supply waits for a trigger on the TTL 5 line before changing the output signal.
	TTL6	9	The power supply waits for a trigger on the TTL 6 line before changing the output signal.
	TTL7	10	The power supply waits for a trigger on the TTL 7 line before changing the output signal.
	ECLO	11	The power supply waits for a trigger on the ECL 0 line before changing the output signal.
	ECL1	12	The power supply waits for a trigger on the ECL 1 line before changing the output signal.
	PXI Star	13	The power supply waits for a trigger on the PXI STAR line before changing the output signal.
	RTSI 0	14	The power supply waits for a trigger on the RTSI 0 line before changing the output signal.
	RTSI 1	15	The power supply waits for a trigger on the RTSI 1 line before changing the output signal.
	RTSI 2	16	The power supply waits for a trigger on the RTSI 2 line before changing the output signal.
	RTSI 3	17	The power supply waits for a trigger on the RTSI 3 line before changing the output signal.
	RTSI 4	18	The power supply waits for a trigger on the RTSI 4 line before changing the output signal.
	RTSI 5	19	The power supply waits for a trigger on

			the RTSI 5 line before changing the output signal.
	RTSI 6	20	The power supply waits for a trigger on the RTSI 6 line before changing the output signal.

Related Topics

[IviDCPwr VI Tree](#)

[IviDCPwr Properties](#)

IviDCPwr Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviDCPwr Errors	0xBFFFA2001 to 0xBFFFA3FFF
IviDCPwr Warnings	0x3FFFA2001 to 0x3FFFA3FFF
IVI Specific Driver Errors	0xBFFFA4000 to 0xBFFFA5FFF
IVI Specific Driver Warnings	0x3FFFA4000 to 0x3FFFA5FFF
IVI Errors	0xBFFFA0000 to 0xBFFFA1FFF
IVI Warnings	0x3FFFA0000 to 0x3FFFA1FFF
Common Instrument Driver Errors	0xBFFFC0000 to 0xBFFFCFFFF
Common Instrument Driver Warnings	0x3FFFC0000 to 0x3FFFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0xFFFFFFF

IviDCPwr Error Code

Error	Value	Message
IVIDCPWR_ERROR_NOT_TRIGGER_SOFTWARE	0xBFFA1001	The trigger source is not software trigger.

Related Topic

[IVI Status Codes](#)

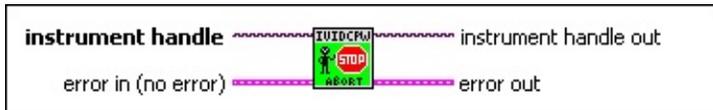
IviDCPwr Functions

Expand this book to view an alphabetized list of IviDCPwr functions.

IviDCPwr Abort

IviDCPwrTrigger Capability Group [TRG]

This VI aborts all pending output changes.



Note If you call this VI after calling the [IviDCPwr Initiate \[TRG\]](#) VI, the power supply ignores any trigger and does not change the output.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



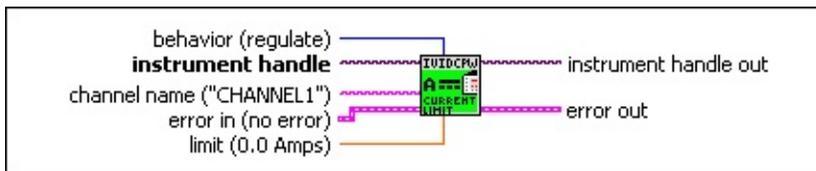
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Current Limit

IviDCPwr Base Capability Group

This VI configures the current limit. You specify the output current limit value and the behavior of the power supply when the output current is greater than or equal to that value.



I/O **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

behavior (regulate) Pass the behavior you want the power supply to exhibit when the output current is greater than or equal to the value of the **limit** parameter. The driver uses this value to set the [IviDCPwr Current Limit Behavior](#) property.

Defined Values:

IVIDCPWR_VAL_CURRENT_REGULATE—Regulatory limit

IVIDCPWR_VAL_CURRENT_TRIP—Trip limit

Default Value: IVIDCPWR_VAL_CURRENT_REGULATE

- limit (0.0 amps)** Pass the current limit you want to use. The driver uses this value to set the [IviDCPwr Current Limit](#) property.

Units: amps

Default Value: 0.0

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

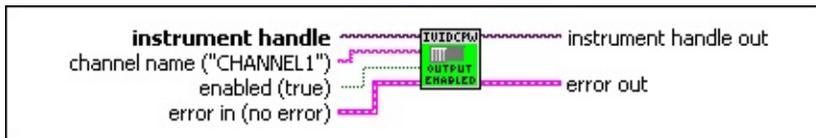
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Output Enabled

IviDCPwr Base Capability Group

Configures whether the signal that the power supply produces on a channel appears at the output connector.



 **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

enabled (true) Pass whether you want the signal the power supply produces on a channel to appear at the output connector. The driver uses this value to set the [IviDCPwr Output Enabled](#) property.

Valid Values:

TRUE - Enable the output

FALSE - Disable the output

Default Value: TRUE

 **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Output Range

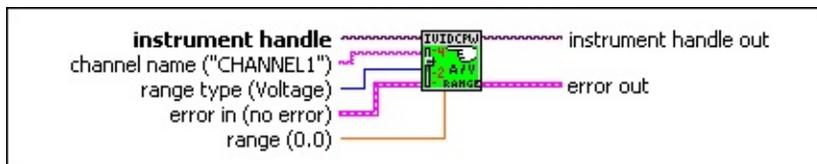
IviDCPwr Base Capability Group

Configures the power supply's output range on a channel. You specify whether you want to configure the voltage or current range, and the value to which to set the range.



Notes

1. Setting a voltage range can invalidate a previously configured current range.
2. Setting a current range can invalidate a previously configured voltage range.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

- range type (Voltage)** Pass the type of range you want to configure.

Defined Values:

IVIDCPWR_VAL_RANGE_VOLTAGE—Voltage Range

IVIDCPWR_VAL_RANGE_CURRENT—Current Range

Default Value: IVIDCPWR_VAL_RANGE_VOLTAGE



Notes

1. Setting a voltage range can invalidate a previously configured current range.
2. Setting a current range can invalidate a previously configured voltage range.

- range (0.0)** Pass the range in which you want the power supply to operate.

If you pass a value of IVIDCPWR_VAL_RANGE_CURRENT for the **RangeType** parameter, specify this parameter in amps.

If you pass a value of IVIDCPWR_VAL_RANGE_VOLTAGE for the **RangeType** parameter, specify this parameter in volts.

Units: volts (for voltage range)
 amps (for current range)

Default Value: 0.0

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

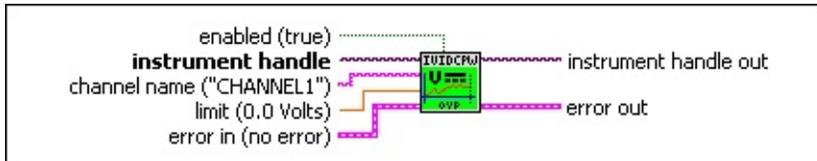
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure OVP

IviDCPwr Base Capability Group

This VI configures the power supply's overvoltage protection. You specify the overvoltage limit and the behavior of the power supply when the output voltage is greater than or equal to that value.

When the enabled parameter is FALSE, the **limit** parameter does not affect the instrument's behavior, and the driver ignores the **limit** parameter.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

enabled (true) Pass whether you want to enable or disable the OVP limit. The driver uses this value to set the [IviDCPwr OVP Enabled](#) property.

Defined Values:

TRUE—Enable OVP limit

FALSE—Disable OVP limit

Default Value: TRUE

- limit (0.0 V)** Pass the overvoltage protection limit you want to use. The driver uses this value to set the [IviDCPwr OVP Limit](#) property.

Units: volts (V)

Default Value: 0.0 V

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

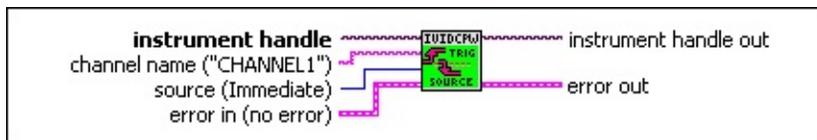
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Trigger Source

IviDCPwrTrigger Capability Group [TRG]

This VI configures the trigger source to which the power supply responds after you call IviDCPwr Initiate [TRG]. When the power supply receives a trigger signal on the source you specify, it changes its current and voltage outputs.

You configure the triggered current and voltage outputs with the IviDCPwr Configure Triggered Current Limit [TRG] and IviDCPwr Configure Triggered Voltage Level [TRG] VIs.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

source (Immediate) Pass the trigger source to which you want the power supply to respond. The driver sets the [IviDCPwr Trigger Source \[TRG\]](#) property to this value.

Defined Values:

IVIDCPWR_VAL_TRIG_IMMEDIATE—The power supply does not wait for a trigger of any kind.

IVIDCPWR_VAL_TRIG_EXTERNAL—The power supply waits for a trigger on the external trigger point.

IVIDCPWR_VAL_SOFTWARE_TRIG—The power supply waits until you call the [IviDCPwr Send Software Trigger \[SWT\]](#) VI.

IVIDCPWR_VAL_TRIG_TTL0—The power supply waits until it receives a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL1—The power supply waits until it receives a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL2—The power supply waits until it receives a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL3—The power supply waits until it receives a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL4—The power supply waits until it receives a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL5—The power supply waits until it receives a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL6—The power supply waits until it receives a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_TTL7—The power supply waits until it receives a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments).

IVIDCPWR_VAL_TRIG_ECL0—The power supply waits until it receives a trigger on the VXI ECL0 line.

IVIDCPWR_VAL_TRIG_ECL1—The power supply waits until it receives a trigger on the VXI ECL1 line.

IVIDCPWR_VAL_TRIG_PXI_STAR—The power supply waits until it receives a trigger on the PXI STAR trigger bus.

IVIDCPWR_VAL_TRIG_RTSI_0—The power supply waits until it receives a trigger on RTSI line 0.

IVIDCPWR_VAL_TRIG_RTSI_1—The power supply waits until it receives a trigger on RTSI line 1.

IVIDCPWR_VAL_TRIG_RTSI_2—The power supply waits until it receives a trigger on RTSI line 2.

IVIDCPWR_VAL_TRIG_RTSI_3—The power supply waits until it receives a trigger on RTSI line 3.

IVIDCPWR_VAL_TRIG_RTSI_4—The power supply waits until it receives a trigger on RTSI line 4.

IVIDCPWR_VAL_TRIG_RTSI_5—The power supply waits until it receives a trigger on RTSI line 5.

IVIDCPWR_VAL_TRIG_RTSI_6—The power supply waits until it receives a trigger on RTSI line 6.

Default Value: IVIDCPWR_VAL_TRIG_IMMEDIATE

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

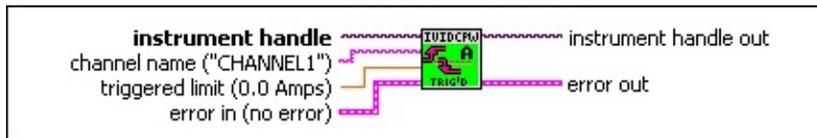
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Triggered Current Limit

IviDCPwrTrigger Capability Group [TRG]

This VI configures the current limit the power supply uses after it receives a trigger.



I/O **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

triggered limit (0.0 amps) Pass the current limit you want the power supply to use after it receives a trigger. The driver sets the [IviDCPwr Triggered Current Limit \[TRG\]](#) property to this value.

Units: amps

Default Value: 0.0 amps

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



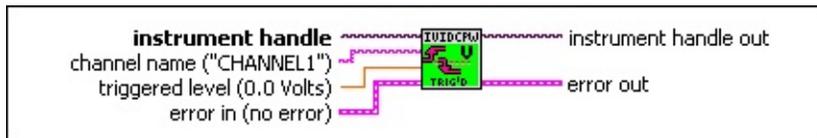
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Triggered Voltage Level

IviDCPwrTrigger Capability Group [TRG]

This VI configures the DC voltage level the power supply attempts to generate after it receives a trigger.



 **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

triggered level (0.0 V) Pass the DC voltage level you want the power supply to attempt to generate after it receives a trigger. The driver sets the [IviDCPwr Triggered Voltage Level \[TRG\]](#) property to this value.

Units: volts (V)

Default Value: 0.0 V

 **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



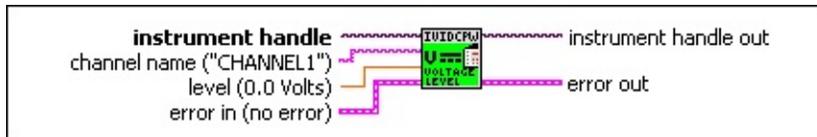
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Configure Voltage Level

IviDCPwr Base Capability Group

This VI configures the DC voltage level that the power supply attempts to generate.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

level (0.0 V) Pass the DC voltage you want the power supply to attempt to generate. The driver sets the [IviDCPwr Voltage Level](#) property to this value.

Units: volts (V)

Default Value: 0.0 V

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



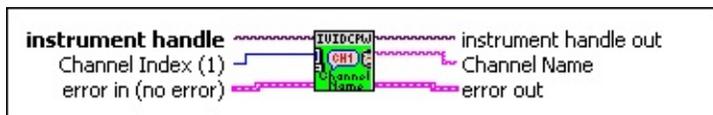
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Get Channel Name

IviDCPwr Base Capability Group

This VI returns the specific driver defined channel name that corresponds to the one-based index that you specify. If the value you pass for the **Index** parameter is less than one or greater than the value of the Channel Count property, the function returns an empty string in the **Name** parameter and returns an error.



- instrument handle** The instrument handle that you obtain from the [Ivi DCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI. The handle identifies a particular instrument session.
- index (1)** A one-based index that defines which name to return. The index must be less than or equal to the [ClassPrefix Channel Count](#) property.
- instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

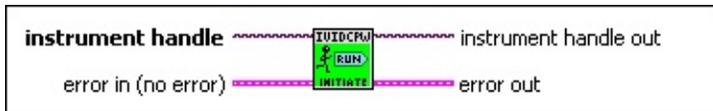
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
- output channel name** Returns the name of the Channel specified by the **index** parameter.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Initiate

IviDCPwrTrigger Capability Group [TRG]

This VI initiates output changes that you previously specified. After you call this VI, the power supply waits for the trigger you specify with the [IviDCPwr Configure Trigger Source \[TRG\]](#) VI. After the power supply detects the trigger, it updates its voltage level and current limit to the values you specify with the [IviDCPwr Configure Triggered Voltage Level \[TRG\]](#) and [IviDCPwr Configure Triggered Current Limit \[TRG\]](#) VIs.



 **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

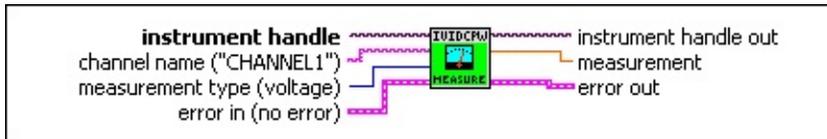
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Measure

IviDCPwrMeasurement [MSR]

This VI takes a single measurement on the channel you specify.



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

measurement type (voltage) Pass the measurement you want the power supply to take.

Defined Values:

IVIDCPWR_VAL_MEASURE_VOLTAGE—DC voltage

IVIDCPWR_VAL_MEASURE_CURRENT—DC current

Default Value: IVIDCPWR_VAL_MEASURE_VOLTAGE

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



measurement Returns the measured value.

If you pass a value of `IVIDCPWR_VAL_MEASURE_CURRENT` for the **measurementType** parameter, this parameter returns a value in amps.

If you pass a value of `IVIDCPWR_VAL_MEASURE_VOLTAGE` for the **measurementType** parameter, this parameter returns a value in volts.

Units: volts (for voltage measurement)
amps (for current measurement)



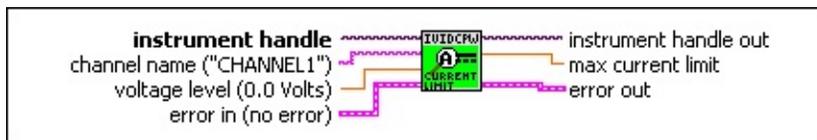
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Query Max Current Limit

IviDCPwr Base Capability Group

This VI returns the maximum programmable current limit that the power supply accepts for a particular voltage level on a channel for the output range to which the power supply is currently configured.



I/O **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

voltage level (0.0 V) Pass the voltage level for which to determine the maximum programmable current limit.

Units: volts (V)

Default Value: 0.0

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



max current limit This parameter returns the maximum programmable current limit for the voltage level you specify.

Units: amps (A)



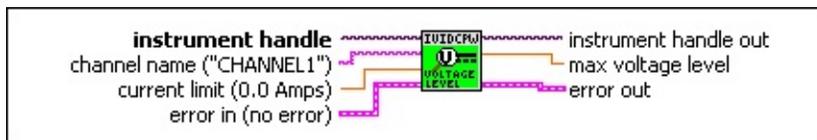
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Query Max Voltage Level

IviDCPwr Base Capability Group

This VI returns the maximum programmable voltage level that the power supply accepts for a particular current limit on a channel for the output range to which the power supply is currently configured.



I/O **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

current limit (0.0 amps) Pass the current limit for which to determine the maximum programmable voltage level.

Units: amps (A).

Default Value: 0.0

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.



max voltage level This parameter returns the maximum programmable voltage level for the current limit you specify.

Units: volts (V)



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

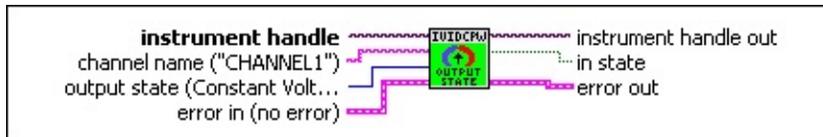
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Query Output State

IviDCPwr Base Capability Group

This VI returns whether the power supply is in a particular output state.

[Details](#)



instrument handle The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

output state (Constant Voltage) Pass the output state for which you want to query.

Defined Values:

IVIDCPWR_VAL_OUTPUT_CONSTANT_VOLTAGE—Constant Voltage State

IVIDCPWR_VAL_OUTPUT_CONSTANT_CURRENT—Constant Current State

IVIDCPWR_VAL_OUTPUT_UNREGULATED—Unregulated State

IVIDCPWR_VAL_OUTPUT_OVER_VOLTAGE—Overvoltage State

IVIDCPWR_VAL_OUTPUT_OVER_CURRENT—Overcurrent State
Default Value: IVIDCPWR_VAL_OUTPUT_CONSTANT_VOLTAGE

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

in state This parameter returns TRUE if the power supply is currently in the state you specify with the **OutputState** parameter, and FALSE if it is not.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Query Output State Details

A constant voltage condition occurs when the output voltage is equal to the value of the [IviDCPwr Voltage Level](#) property and the current is less than or equal to the value of the IviDCPwr [Current Limit](#) property.

A constant current condition occurs when the output current is equal to the value of the IviDCPwr Current Limit property and the [IviDCPwr Current Limit Behavior](#) property is set to IVIDCPWR_VAL_CURRENT_REGULATE.

An unregulated condition occurs when the output voltage is less than the value of the [IviDCPwr Voltage Level](#) property and the current is less than the value of the IviDCPwr Current Limit property.

An overvoltage condition occurs when the output voltage is equal to or greater than the value of the [IviDCPwr OVP Limit](#) property and the [IviDCPwr OVP Enabled](#) property is set to TRUE.

An over-current condition occurs when the output current is equal to or

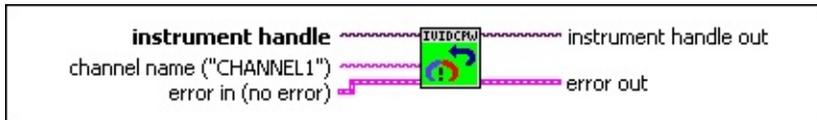
greater than the value of the [IviDCPwr Current Limit](#) property and the [IviDCPwr Current Limit Behavior](#) property is set to IVIDCPWR_VAL_CURRENT_TRIP.

When either an overvoltage condition or an over-current condition occurs, the power supply's output protection disables the output. If the power supply is in an overvoltage or over-current state, it does not produce power until the output protection is reset. The [IviDCPwr Reset Output Protection](#) VI resets the output protection. Once the output protection is reset, the power supply resumes generating a power signal.

IviDCPwr Reset Output Protection

IviDCPwr Base Capability Group

This VI clears all output-protection conditions on the power supply.



 **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

channel name ("CHANNEL 1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: CHANNEL 1

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

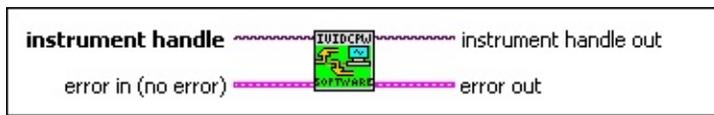
The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

IviDCPwr Send Software Trigger

IviDCPwrSoftwareTrigger [SWT]

This VI sends a command to trigger the power supply. Call this VI if you configure the power supply to respond to software triggers. If the power supply is not configured to respond to software triggers, this VI returns the error IVIDCPWR_ERROR_TRIGGER_NOT_SOFTWARE (0xBFFA1001).



 **instrument handle** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDCPwr Initialize](#) or [IviDCPwr Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDCPwr Properties

Expand this book to view an alphabetized list of IviDCPwr properties.

Active Channel

Active Channel Property

Data Type	Access	Applies to	Coercion	High Level VIs
abc	WO	n/a	none	none

Description

Specifies the channel name used to access all subsequent channel-based properties in this property node. Set the channel before setting channel-based properties. Pass a name that the instrument driver defines or a virtual channel name the end-user defines in the IVI configuration file in MAX.

Property Node Path

<classdriver>»Active Channel

IviDCPwr Channel Count

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of channels available on the instrument.

Property Node Path

IviDCPwr»Inherent IVI Settings»Specific Driver Capabilities»Channel Count

IviDCPwr Current Limit

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
	R/W	Channel	N/A	IviDCPwr Configure Current Limit

Description

This channel-based property specifies the output current limit. The units are Amperes.

The value of the [IviDCPwr Current Limit Behavior](#) property determines the behavior of the power supply when the output current is equal to or greater than the value of this property.

Property Node Path

IviDCPwr»Basic Operation»Current Limit

IviDCPwr Current Limit Behavior

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviDCPwr Configure Current Limit

Description

This channel-based property specifies the behavior of the power supply when the output current is equal to or greater than the value of the [IviDCPwr Current Limit](#) property.

Property Node Path

IviDCPwr»Basic Operation»Current Limit Behavior

Defined Values

[IVIDCPWR_VAL_CURRENT_REGULATE](#)

[IVIDCPWR_VAL_CURRENT_TRIP](#)

IviDCPwr Output Enabled

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviDCPwr Configure Output Enabled

Description

This channel-based property specifies whether the signal the power supply produces appears at the output connector.

Property Node Path

IviDCPwr»Basic Operation»Output Enabled

IviDCPwr OVP Enabled

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviDCPwr Configure OVP

Description

This channel-based property specifies whether the power supply provides overvoltage protection. If this property is set to TRUE, the power supply disables the output when the output voltage is greater than or equal to the value of the [IviDCPwr OVP Limit](#) property.

Property Node Path

IviDCPwr»Basic Operation»OVP Enabled

IviDCPwr OVP Limit

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
	R/W	Channel	N/A	IviDCPwr Configure OVP

Description

This channel-based property specifies the voltage the power supply allows. The units are volts.

If the [IviDCPwr OVP Enabled](#) property is set to TRUE, the power supply disables the output when the output voltage is greater than or equal to the value of this property.

If the IviDCPwr OVP Enabled property is set to FALSE, this property does not affect the behavior of the instrument.

Property Node Path

IviDCPwr»Basic Operation»OVP Limit

IviDCPwr Trigger Source [TRG]

IviDCPwrTrigger Property [TRG]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviDCPwr Configure Trigger Source [TRG]

Description

This channel-based property specifies the trigger source. After you call [IviDCPwr Initiate \[TRG\]](#), the power supply waits for a trigger event from the source you specify with this property. After a trigger event occurs, the power supply changes the voltage level to the value of the [IviDCPwr Triggered Voltage Level \[TRG\]](#), and the current limit to the value of the [IviDCPwr Triggered Current Limit \[TRG\]](#), properties.

Property Node Path

IviDCPwr»Trigger»Trigger Source [TRG]

Defined Values

[IVIDCPWR_VAL_TRIG_IMMEDIATE](#)

[IVIDCPWR_VAL_TRIG_EXTERNAL](#)

[IVIDCPWR_VAL_SOFTWARE_TRIG](#)

[IVIDCPWR_VAL_TRIG_TTL0](#)

[IVIDCPWR_VAL_TRIG_TTL1](#)

[IVIDCPWR_VAL_TRIG_TTL2](#)

[IVIDCPWR_VAL_TRIG_TTL3](#)

[IVIDCPWR_VAL_TRIG_TTL4](#)

[IVIDCPWR_VAL_TRIG_TTL5](#)

[IVIDCPWR_VAL_TRIG_TTL6](#)

[IVIDCPWR_VAL_TRIG_TTL7](#)

[IVIDCPWR_VAL_TRIG_ECL0](#)

[IVIDCPWR_VAL_TRIG_ECL1](#)

[IVIDCPWR_VAL_TRIG_PXI_STAR](#)

IVIDCPWR_VAL_TRIG_RTSI_0

IVIDCPWR_VAL_TRIG_RTSI_1

IVIDCPWR_VAL_TRIG_RTSI_2

IVIDCPWR_VAL_TRIG_RTSI_3

IVIDCPWR_VAL_TRIG_RTSI_4

IVIDCPWR_VAL_TRIG_RTSI_5

IVIDCPWR_VAL_TRIG_RTSI_6

IviDCPwr Triggered Current Limit [TRG]

IviDCPwrTrigger Property [TRG]

Data Type	Access	Applies to	Coercion	High Level VIs
	R/W	Channel	N/A	IviDCPwr Configure Triggered Current Limit [TRG]

Description

This channel-based property specifies the value to which the power supply sets the current limit after a trigger event occurs. The units are amps.

After you call [IviDCPwr Initiate \[TRG\]](#), the power supply waits for a trigger event from the source you specify with the [IviDCPwr Trigger Source \[TRG\]](#), property. After a trigger event occurs, the power supply sets the current limit to the value of this property. After a trigger occurs, the value of the [IviDCPwr Current Limit \[TRG\]](#) property reflects the new value to which the current limit has been set.

Property Node Path

IviDCPwr»Trigger»Triggered Current [TRG]

IviDCPwr Triggered Voltage Level [TRG]

IviDCPwrTrigger Property

Data Type	Access	Applies to	Coercion	High Level VIs
	R/W	Channel	N/A	IviDCPwr Configure Triggered Voltage Level [TRG]

Description

This channel-based property specifies the value to which the power supply sets the voltage level after a trigger event occurs. The units are volts.

After you call [IviDCPwr Initiate \[TRG\]](#), the power supply waits for a trigger event from the source you specify with the [IviDCPwr Trigger Source \[TRG\]](#), property. After a trigger event occurs, the power supply sets the voltage level to the value of this property.

After a trigger occurs, the value of the [IviDCPwr Voltage Level \[TRG\]](#) property reflects the new value to which the voltage level has been set.

Property Node Path

IviDCPwr»Trigger»Triggered Voltage Level [TRG]

IviDCPwr Voltage Level

IviDCPwr Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
	R/W	Channel	N/A	IviDCPwr Configure Voltage Level

Description

This channel-based property specifies the voltage level the DC power supply attempts to generate. The units are volts.

Property Node Path

IviDCPwr»Basic Operation»Voltage Level

IviDCPwr Defined Values

Expand this book to view an alphabetized list of IviDCPwr defined values.

IVIDCPWR_VAL_CURRENT_REGULATE

The power supply regulates the output current at the value of the IviDCPwr Current Limit property when the output reaches or exceeds that value.

Defined Value: 0

IVIDCPWR_VAL_CURRENT_TRIP

The power supply disables the output when the output current reaches or exceeds the value of the IviDCPwr Current Limit property.

Defined Value: 1

IVIDCPWR_VAL_SOFTWARE_TRIG

The power supply waits for the [IviDCPwr Send Software Trigger](#) VI to execute before changing the output signal.

Defined Value: 2

IVIDCPWR_VAL_TRIG_ECL0

The power supply waits for a trigger on the ECL 0 line before changing the output signal.

Defined Value: 11

IVIDCPWR_VAL_TRIG_ECL1

The power supply waits for a trigger on the ECL 1 line before changing the output signal.

Defined Value: 12

IVIDCPWR_VAL_TRIG_EXTERNAL

The power supply waits for an external trigger before changing the output signal.

Defined Value: 1

IVIDCPWR_VAL_TRIG_IMMEDIATE

The power supply does not wait for a trigger before changing the output signal.

Defined Value: 0

IVIDCPWR_VAL_TRIG_IMMEDIATE

The power supply does not wait for a trigger before changing the output signal.

Defined Value: 0

IVIDCPWR_VAL_TRIG_PXI_STAR

The power supply waits for a trigger on the PXI STAR line before changing the output signal.

Defined Value: 13

IVIDCPWR_VAL_TRIG_RTSI_0

The power supply waits for a trigger on the RTSI 0 line before changing the output signal.

Defined Value: 14

IVIDCPWR_VAL_TRIG_RTSI_1

The power supply waits for a trigger on the RTSI 1 line before changing the output signal.

Defined Value: 15

IVIDCPWR_VAL_TRIG_RTSI_2

The power supply waits for a trigger on the RTSI 2 line before changing the output signal.

Defined Value: 16

IVIDCPWR_VAL_TRIG_RTSI_3

The power supply waits for a trigger on the RTSI 3 line before changing the output signal.

Defined Value: 17

IVIDCPWR_VAL_TRIG_RTSI_4

The power supply waits for a trigger on the RTSI 4 line before changing the output signal.

Defined Value: 18

IVIDCPWR_VAL_TRIG_RTSI_5

The power supply waits for a trigger on the RTSI 5 line before changing the output signal.

Defined Value: 19

IVIDCPWR_VAL_TRIG_RTSI_6

The power supply waits for a trigger on the RTSI 6 line before changing the output signal.

Defined Value: 20

IVIDCPWR_VAL_TRIG_TTL0

The power supply waits for a trigger on the TTL 0 line before changing the output signal.

Defined Value: 3

IVIDCPWR_VAL_TRIG_TTL1

The power supply waits for a trigger on the TTL 1 line before changing the output signal.

Defined Value: 4

IVIDCPWR_VAL_TRIG_TTL2

The power supply waits for a trigger on the TTL 2 line before changing the output signal.

Defined Value: 5

IVIDCPWR_VAL_TRIG_TTL3

The power supply waits for a trigger on the TTL 3 line before changing the output signal.

Defined Value: 6

IVIDCPWR_VAL_TRIG_TTL4

The power supply waits for a trigger on the TTL 4 line before changing the output signal.

Defined Value: 7

IVIDCPWR_VAL_TRIG_TTL5

The power supply waits for a trigger on the TTL 5 line before changing the output signal.

Defined Value: 8

IVIDCPWR_VAL_TRIG_TTL6

The power supply waits for a trigger on the TTL 6 line before changing the output signal.

Defined Value: 9

IVIDCPWR_VAL_TRIG_TTL7

The power supply waits for a trigger on the TTL 7 line before changing the output signal.

Defined Value: 10

Properties for Controlling IviDCPwr Simulation



Note You can set Simulation Attributes in MAX on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** or in the simulation interactive panels. Refer to **National Instruments IVI Driver Help»Configuring Your System** for more information.

Two sets of properties exist for use with the IviDCPwr Simulation Driver. The following table describes properties that control behavior of the driver. Another table lists [properties that simulate the status](#) of specific driver VIs.

An [IviDCPwr Simulator Setup Dialog Box](#) exists to help you configure the measurement simulation for the IviDCPwr Simulator driver.

IviDCPwr Properties for Controlling Simulation

Name	Data Type	Access	Applies to	Description
INTERACTIVE_SIMULATION	Boolean	W	No	Specifies whether to set the driver to interactive mode. In interactive mode, the simulation driver uses pop-up user interface panels for getting information from you. Non-interactive mode is useful when you do not want the interactive panels to interrupt your test program.
SELF_TEST_CODE	Integer	W	No	Specifies the self-test code. When the IviDCPwr Self-Test VI is called, the self-test result parameter returns this

				value.
SELF_TEST_MSG	String	W	No	Specifies the self-test message. When the IviDCPwr Self-Test VI is called, the self-test message parameter returns this string.
ERROR_QUERY_CODE	Integer	W	No	Specifies the error-query code. When the IviDCPwr Error-Query VI is called, the error code parameter returns this value.
ERROR_QUERY_MSG	String	W	No	Specifies the error-query message. When the IviDCPwr Error-Query VI is called, the error message parameter returns this string.
DRIVER_REV_QUERY	String	W	No	Specifies the instrument

				driver revision message. When the IviDCPwr Revision QueryVI is called, the instrument driver revision parameter returns this string.
INSTR_REV_QUERY	String	W	No	Specifies the firmware revision message. When the IviDCPwr Revision Query VI is called, the firmware revision parameter returns this string.
SIMULATE_STATUS_CODE	Boolean	W	No	Specifies whether to simulate return codes from the IviDCPwr driver VIs. Refer to the Properties for IviDCPwr Simulation

			table for more
--	--	--	----------------

Related Topics

[Setting Up Simulated Errors](#)

[Setting Up Simulation for VXI*plug&play* VIs](#)

IviDCPwr Simulator Setup Dialog Box

Two sets of properties exist for use with the IviDCPwr Simulation Driver. You use the dialog box shown below to configure the measurement simulation for the IviDCPwr Simulator driver. Scroll down to see a description of each control.



Simulator Setup Dialog Box

Always prompt for output data simulation

Leaving this control selected causes the panel to appear at every measurement acquisition event in your program. All simulated acquisitions in the program use the same measurement configuration information. Unselecting this control causes the measurement simulation panel to never appear again during the course of your program.

Channel

This control specifies the channel for which you want to configure a simulation measurement.

Measurement

Enter the measurement base to use for all computer generated measurements. All computer generated measurement values fall within the range of this value plus or minus the value of the Noise control. Valid Range: any value

Noise

Enter the reading noise to use for all computer generated measurements. All computer generated measurement values fall within the range of the value of the Measurement control plus or minus this value. Valid Range: any positive value

Properties for Status Simulation in IviDCPwr

When an IviDCPwr driver function is called, the driver queries whether NISDCPWR_ATTR_SIMULATE_STATUS_CODE is enabled. If enabled, the driver gets the appropriate function's simulation status code. If the value of the simulation status code represents a warning (has a positive value), then it is returned as the function's return status only if no other error or warning occurred before the function ends. If the value of the simulation status code represents an error (has a negative value), then it is returned as the function's return status only if no other error occurred before the function ends.

The following IviDCPwr VIs do not support status code simulation:

- IviDCPwr Get Error
- IviDCPwr Clear Error
- IviDCPwr Get Next Interchange Warning
- IviDCPwr Reset Next Interchange Warning
- IviDCPwr Clear Next Interchange Warning
- IviDCPwr Get Next Coercion Record

Each VI supported by the IviDCPwr class driver has a corresponding property that determines the status code to return when status code simulation is enabled. The following table lists the status code properties and the VI for which they return a value. These properties are all of type ViInt32, non-readable, and non-channel based.



Note In the following table, the literal string NISDCPWR precedes all property names.

IviDCPwr Properties for Status Simulation

Property	VI
INIT_STATUS	IviDCPwr Initialize, IviDCPwr Initialize With Options
CLOSE_STATUS	IviDCPwr Close
RESET_STATUS	IviDCPwr Reset
SELF_TEST_STATUS	IviDCPwr Self-Test
ERROR_QUERY_STATUS	IviDCPwr Error-Query
ERROR_MESSAGE_STATUS	IviDCPwr Error Message
REVISION_QUERY_STATUS	IviDCPwr Revision Query
RESET_DEFAULT_STATUS	IviDCPwr Reset With Defaults
DISABLE_STATUS	IviDCPwr Disable
INVALIDATE_STATUS	IviDCPwr Invalidate All Attributes
GET_CH_NAME_STATUS	IviDCPwr Get Channel Name
CONFIGURE_OUTPUT_ENABLED_STATUS	IviDCPwr Configure Output Enabled
CONFIGURE_OUTPUT_RANGE_STATUS	IviDCPwr Configure Output Range
CONFIGURE_CURRENT_LIMIT_STATUS	IviDCPwr Configure Current Limit
CONFIGURE_OVP_STATUS	IviDCPwr

	Configure OVP
CONFIGURE_VOLTAGE_LEVEL_STATUS	IviDCPwr Configure Voltage Level
QUERY_OUTPUT_STATE_STATUS	IviDCPwr Query Output State
QUERY_MAX_CURRENT_LIMIT_STATUS	IviDCPwr Query Max Current Lim
QUERY_MAX_VOLTAGE_LEVEL_STATUS	IviDCPwr Query Max Voltage Lev
RESET_OUTPUT_PROTECTION_STATUS	IviDCPwr Reset Output Protectio
CONFIGURE_TRIGGER_SOURCE_STATUS	IviDCPwr Configure Trigge Source
CONFIGURE_TRIGGERED_VOLTAGE_LEVEL_STATUS	IviDCPwr Configure Triggered Voltage Level
CONFIGURE_TRIGGERED_CURRENT_LIMIT_STATUS	IviDCPwr Configure Triggered Current Limit
ABORT_STATUS	IviDCPwr Abort
INITIATE_STATUS	IviDCPwr Initiate
SEND_SOFTWARE_TRIGGER_STATUS	IviDCPwr Send Software Trigger
MEASURE_STATUS	IviDCPwr Measu

□ IviDmm Class Driver Reference for LabVIEW

Concepts

[IviDmm Class Driver Overview](#)

[IviDmm Extension Capability Groups](#)

[IviDmm Behavior Models](#)

IviDmm VI and Property Reference

[IviDmm VI Tree](#)

[IviDmm Properties by Group](#)

[IviDmm Property Value Definitions](#)

[IviDmm Error and Completion Codes](#)

Other Reference

[Interchangeability Checking](#)

[Simulation](#)

IviDmm Class Driver Overview

This instrument driver provides programming support for the IviDmm Class. The IviDmm class is designed to support the typical DMM as well as common extended functionality found in more complex instruments. The IviDmm class driver conceptualizes a DMM as an instrument that can measure scalar quantities of an input signal and can be applied to a wide variety of instruments. Typically the measured quantity is a voltage (AC and DC), current, or resistance. However, the IviDmm class driver can support instruments that measure other quantities such as temperature and frequency. The driver contains all the VIs that the IVI Foundation requires for the IviDmm specification. This driver requires NI-VISA and the IVI Compliance Package to be installed.

The IviDmm class is divided into a [IviDmmBase capability group](#) and several extension groups. The base capability group is used to configure a DMM for a typical measurement (which includes setting the measurement VI, desired range, desired resolution, and trigger source), initiating that measurement, and returning a measured value.

Many DMMs support measurement types that require additional parameters to be configured, such as the minimum and maximum frequency of the input signal for AC measurements. The IviDmm class defines extension groups for each measurement type that requires these additional parameters.

The IviDmm class also defines an extension group called IviDmmMultiPoint. The IviDmmMultiPoint extension group is used to configure DMMs that can acquire multiple measurements based on multiple triggers and take multiple measurements per trigger. This type of instrument used in conjunction with a scanner is typically used to implement a scanning DMM.

In addition, the IviDmm class defines extension groups that configure advanced settings such as auto-zero and powerline frequency, or return additional information about the current state of the instrument such as aperture time.

Use this driver to develop programs that are independent of a particular DMM. You can use this class driver with any DMM that has an IVI instrument specific driver that is compliant with the IviDmm class. The IviDmm class driver accesses the specific driver for your instrument using

the configuration information you supply with MAX.

Extension Capability Groups

The IviDmm class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Abbreviation	Description
IviDmmACMeasurement	[AC]	Defines properties that configure additional settings for AC measurements. These properties are the minimum and maximum frequency components of the input signal. This extension group also defines VIs that configure these properties.
IviDmmAutoRangeValue	[ARV]	Supports DMMs with the capability to return the actual range value when auto ranging.
IviDmmAutoZero	[AZ]	Supports DMMs with the capability to take an auto zero reading. In general, the auto-zero capability of a DMM normalizes all measurements based on a Zero Reading.
IviDmmDeviceInfo	[DI]	Defines a VI and a set of properties that you can query to gain

		additional information about the instrument's configuration
<u>IviDmmFrequencyMeasurement</u>	[FREQ]	Supports DMMs that take frequency measurements. It defines properties that are required to configure additional parameters needed for frequency measurements.
<u>IviDmmMultiPoint</u>	[MP]	Supports instruments capable of acquiring measurements based on multiple triggers, and acquiring multiple measurements for each trigger
<u>IviDmmPowerLineFrequency</u>	[PLF]	Supports DMMs with the capability to specify the powerline frequency.
<u>IviDmmResistanceTemperature Device</u>	[RTD]	Supports DMMs that take temperature measurements using a resistance temperature device (RTD) transducer type.
<u>IviDmmSoftwareTrigger</u>	[SWT]	Supports DMMs that can initiate a measurement based on a software trigger signal. You can send a software trigger to cause the DMM to

		initiate a measurement.
IviDmmTemperatureMeasurement	[TMP]	Supports DMMs that take temperature measurements with a thermocouple, an RTD, or a thermistor transducer type. This extension group selects the transducer type. Other capability groups further configure temperature settings based on the transducer type.
IviDmmThermistor	[THM]	Supports DMMs that take temperature measurements using a thermistor transducer type.
IviDmmThermocouple	[TC]	Supports DMMs that take temperature measurements using a thermocouple transducer type.
IviDmmTriggerSlope	[TS]	Supports DMMs that can specify the polarity of the external trigger signal. It defines an property and a VI to configure this polarity.

Related Topics

[IVI Inherent Class Capabilities](#)

IviDmm Extension Capability Groups

The IviDmm class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviDmmACMeasurement</u>	Defines properties that configure additional settings for AC measurements. These properties are the minimum and maximum frequency components of the input signal. This extension group also defines VIs that configure these properties.
<u>IviDmmAutoRangeValue</u>	Supports DMMs with the capability to return the actual range value when auto ranging.
<u>IviDmmAutoZero</u>	Supports DMMs with the capability to take an auto zero reading. In general, the auto-zero capability of a DMM normalizes all measurements based on a Zero Reading.
<u>IviDmmDeviceInfo</u>	Defines a VI and a set of properties that you can query to gain additional information about the instrument's configuration
<u>IviDmmFrequencyMeasurement</u>	Supports DMMs that take frequency measurements. It defines properties that are required to configure additional parameters needed for frequency measurements.
<u>IviDmmMultiPoint</u>	Supports instruments capable of acquiring measurements based on

	multiple triggers, and acquiring multiple measurements for each trigger
<u>IviDmmPowerLineFrequency</u>	Supports DMMs with the capability to specify the powerline frequency.
<u>IviDmmResistanceTemperatureDevice</u>	Supports DMMs that take temperature measurements using a resistance temperature device (RTD) transducer type.
<u>IviDmmSoftwareTrigger</u>	Supports DMMs that can initiate a measurement based on a software trigger signal. You can send a software trigger to cause the DMM to initiate a measurement.
<u>IviDmmTemperatureMeasurement</u>	Supports DMMs that take temperature measurements with a thermocouple, an RTD, or a thermistor transducer type. This extension group selects the transducer type. Other capability groups further configure temperature settings based on the transducer type.
<u>IviDmmThermistor</u>	Supports DMMs that take temperature measurements using a thermistor transducer type.
<u>IviDmmThermocouple</u>	Supports DMMs that take temperature measurements using a thermocouple transducer type.
<u>IviDmmTriggerSlope</u>	Supports DMMs that can specify the polarity of the external trigger signal. It defines an property and a VI to configure this polarity.

IviDmmACMeasurement Extension Group

The IviDmmACMeasurement extension group supports DMMs that take AC voltage or AC current measurements. It defines properties that configure additional settings for AC measurements. These properties are the minimum and maximum frequency components of the input signal. This extension group also defines VIs that configure these properties.

ACMeasurement Extension VI

[IviDmm Configure AC Bandwidth](#)

ACMeasurement Extension Properties

[IviDmm AC Minimum Frequency](#)

[IviDmm AC Maximum Frequency](#)

Related Topics

[IviDmm Behavior Model](#)

[IviDmm VI Tree](#)

[IviDmm Properties](#)

IviDmmAutoRangeValue Extension Group

The IviDmmAutoRangeValue extension supports DMMs with the capability to return the actual range value when auto ranging.

AutoRangeValue Extension VI

[IviDmm Get Auto Range Value](#)

AutoRangeValue Extension Property

[IviDmm Auto Range Value](#)

Related Topics

[IviDmm behavior model](#)

IviDmmAutoZero Extension Group

The IviDmmAutoZero extension group supports DMMs that have the capability to take an auto-zero reading. In general, the auto-zero capability of a DMM normalizes all measurements based on a Zero Reading.

AutoZero Extension VI

[IviDmm Configure Auto Zero Mode](#)

AutoZero Extension Property

[IviDmm Auto Zero](#)

Related Topics

[IviDmm behavior model](#)

IviDmm Base Capability Group

The IviDmm base capability group supports DMMs that take one measurement at a time. The IviDmm base capability group defines properties and their values to configure the type of measurement and how the measurement is performed. These properties include the measurement function, range, resolution, trigger source, and trigger delay. The IviDmm base capability group includes VIs for configuring the DMM as well as initiating and retrieving measurements.

IviDmmBase VIs

[IviDmm Configure Measurement](#)

[IviDmm Configure Trigger](#)

[IviDmm Read](#)

[IviDmm Abort](#)

[IviDmm Initiate](#)

[IviDmm Fetch](#)

[IviDmm Send Software Trigger](#)

[IviDmm Is Over-Range](#)

IviDmmBase Properties

[IviDmm Function](#)

[IviDmm Range](#)

[IviDmm Resolution](#)

[IviDmm Trigger Source](#)

[IviDmm Trigger Delay](#)

Related Topics

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

[IviDmm Behavior Model](#)

For more VI and property information:

[IviDmm VI Tree](#)

[IviDmm Properties](#)

IviDmmDeviceInfo Extension Group

The IviDmmDeviceInfo capability group defines a set of read-only properties that can be queried to gain additional information concerning how the DMM is presently configured.

IviDmmDeviceInfo VI

[IviDmm Get Aperture Time Info](#)

IviDmmDeviceInfo Properties

[IviDmm Aperture Time \[DI\]](#)

[IviDmm Aperture Time Units \[DI\]](#)

Behavior Model

[IviDmm behavior model](#)

IviDmmFrequencyMeasurement Extension Group

The IviDmmFrequencyMeasurement extension group supports DMMs that take frequency measurements. This group defines properties that are required to configure additional parameters needed for frequency measurements.

FrequencyMeasurement Extension VI

[IviDmm Configure Frequency Voltage Range \[FRQ\]](#)

FrequencyMeasurement Extension Property

[IviDmm Frequency Voltage Range](#)

Related Topics

[IviDmm behavior model](#)

IviDmmMultiPoint Extension Group

The IviDmmMultiPoint extension group defines extensions for DMMs capable of acquiring measurements based on multiple triggers, and acquiring multiple measurements for each trigger.

The IviDmmMultiPoint extension group defines additional properties such as sample count, sample trigger, trigger count, and sample interval to control multipoint DMMs. The IviDmmMultiPoint extension group also adds VIs for configuring the DMM as well as starting acquisitions and retrieving multiple measured values.

Multipoint Extension VIs

[IviDmm Configure Meas Complete Dest](#)

[IviDmm Configure Multipoint](#)

[IviDmm Read Multipoint](#)

[IviDmm Fetch Multiple Point](#)

Multipoint Extension Properties

[IviDmm Meas Complete Destination \[MP\]](#)

[IviDmm Sample Count \[MP\]](#)

[IviDmm Sample Interval \[MP\]](#)

[IviDmm Sample Trigger \[MP\]](#)

[IviDmm Trigger Count \[MP\]](#)

Related Topics

[IviDmm Properties](#)

[IviDmm VIs](#)

[IviDmm behavior model](#)

IviDmmPowerLineFrequency Extension Group

The IviDmmPowerLineFrequency extension group supports DMMs with the capability to specify the powerline frequency.

PowerLineFrequency Extension VI

[IviDmm Configure Power Line Frequency \[PLF\]](#)

PowerLineFrequency Extension Property

[IviDmm Powerline Frequency](#)

Related Topics

[IviDmm behavior model](#)

IviDmmResistanceTemperatureDevice Extension Group

The IviDmmResistanceTemperatureDevice extension group supports DMMs that take temperature measurements using a resistance temperature device (RTD) transducer type.

The IviDmm class assumes that you are using a Platinum Resistance Temperature Device.

ResistanceTemperatureDevice Extension VI

[IviDmm Configure RTD](#)

ResistanceTemperatureDevice Extension Properties

[IviDmm RTD Alpha](#)

[IviDmm RTD Resistance](#)

Related Topics

[IviDmm behavior model](#)

IviDmmSoftwareTrigger Extension Group

The IviDmmSoftwareTrigger extension group supports DMMs that can initiate a measurement based on a software trigger signal. You can send a software trigger to cause the DMM to initiate a measurement.

SoftwareTrigger Extension VI

[IviDmm Send Software Trigger](#)

Related Topics

[IviDmm behavior model](#)

IviDmmTemperatureMeasurement Extension Group

The IviDmmTemperatureMeasurement extension group supports DMMs that take temperature measurements with a thermocouple, an RTD, or a thermistor transducer type. This extension group selects the transducer type. Other capability groups further configure temperature settings based on the transducer type.

TemperatureMeasurement Extension VI

[IviDmm Configure Transducer Type](#)

TemperatureMeasurement Extension Property

[IviDmm Transducer Type \[TMP\]](#)

Related Topics

[IviDmm behavior model](#)

IviDmmThermistor Extension Group

The IviDmmThermistor extension group supports DMMs that take temperature measurements using a thermistor transducer type.

The IviDmm class assumes that you are using an interchangeable thermistor. Interchangeable thermistors are thermistors that exhibit similar behavior for a given resistance value.

Thermistor Extension VI

[lviDmm Configure Thermistor \[THM\]](#)

Thermistor Extension Property

[IviDmm Thermistor Resistance](#)

Related Topics

[IviDmm behavior model](#)

IviDmmThermocouple Extension Group

The IviDmmThermocouple extension group supports DMMs that take temperature measurements using a thermocouple transducer type.

Thermocouple Extension VIs

[IviDmm Configure Thermocouple](#)

[IviDmm Configure Fixed Ref Junction](#)

Thermocouple Extension Properties

[IviDmm Fixed Reference Junction](#)

[IviDmm Reference Junction Type](#)

[IviDmm Thermocouple Type](#)

Related Topics

[IviDmm behavior model](#)

IviDmmTriggerSlope Extension Group

The IviDmmTriggerSlope extension group supports DMMs that can specify the polarity of the external trigger signal. It defines an property and a VI to configure this polarity.

TriggerSlope Extension VI

[IviDmm Configure Trigger Slope](#)

TriggerSlope Extension Property

[IviDmm Trigger Slope](#)

Related Topics

[IviDmm behavior model](#)

IviDmm Behavior Models

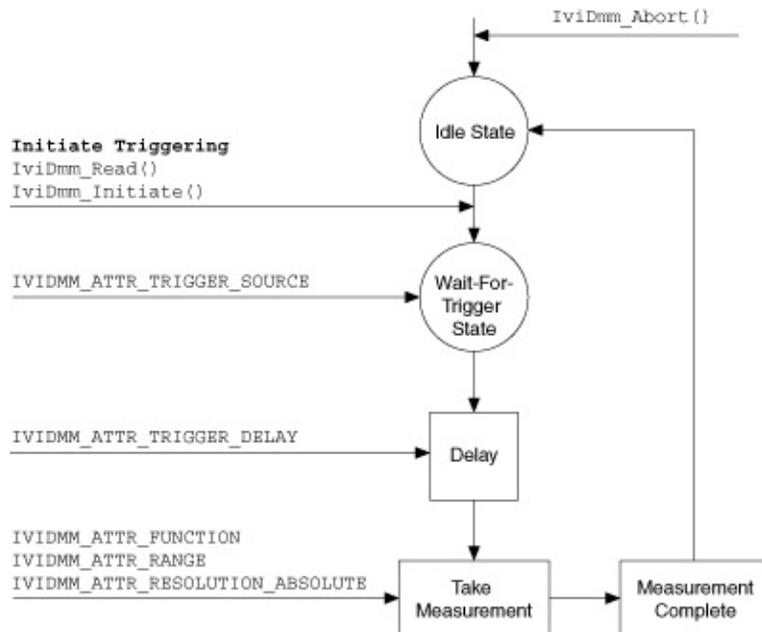
[IviDmm](#)

[IviDmmMultiPoint](#)

[IviDmmSoftwareTrigger](#)

IviDmm Behavior Model

The following state diagram shows relationships between the IviDmm base capability group and DMM behavior.



IviDmm Behavior Model

The main state in the IviDmm Class is the Idle state. The DMM enters the Idle state when it is powered-on, when it successfully completes a measurement, or when you call [IviDmm Abort](#) from a previous measurement. Typically, you configure the DMM while it is in the Idle state. IviDmm properties can be configured individually with the property node or with the high-level [IviDmm Configure Measurement VI](#).

The [IviDmm Read](#) and [IviDmm Initiate](#) VIs cause the DMM to leave the Idle state and transition to the Wait-For-Trigger state. The IviDmm Read VI does not return until the measurement process is complete and the DMM has returned to the Idle state. The IviDmm Initiate VI returns as soon as the DMM leaves the Idle state.

The DMM leaves the Wait-For-Trigger state when it receives a trigger event. The type of trigger event is specified by the property [IviDmm Trigger Source](#).

After the specified trigger event occurs, the DMM waits the amount of time specified by the property [IviDmm Trigger Delay](#) and then takes a measurement. The type of measurement is specified by the [IviDmm](#)

[Function](#), [IviDmm Range](#), and [IviDmm Resolution](#) properties.

If you set the IviDmm Function property to a value that requires an extension capability group, the properties of that capability group further configure the measurement.

After the measurement is taken, the DMM (if it is capable of doing so) generates the Measurement Complete signal and returns to the Idle state.

The IviDmm base capability group does not require that a DMM be able to generate a Measurement Complete signal. The Measurement Complete signal is presented in the Base Capabilities State Diagram to define when the signal is generated as most DMMs generate this signal but may not be able to configure it.

The [IviDmm Fetch](#) VI is used to retrieve measurements that were initiated by the [IviDmm Initiate](#) VI. The measurement data returned from the [IviDmm Read](#) and [IviDmm Fetch](#) VIs is acquired after the DMM has left the Wait-For-Trigger state.

IviDmmMultiPoint Behavior Model

The following state diagram shows relationships between IviDmmMultiPoint extension group and DMM behavior. Only the elements added by the IviDmmMultiPoint capability group are represented in this state diagram.



IviDmmMultiPoint Behavior Model

The IviDmmMultiPoint behavior model builds upon the fundamental IviDmm behavior model and only documents additional items introduced by the IviDmmMultiPoint extension group. The main state is the Idle state. Typically, you configure the IviDmmMultiPoint properties while DMM is in the Idle state. IviDmmMultiPoint properties can be configured individually with the high-level [IviDmm Configure MultiPoint](#) VI.

The [IviDmm Initiate](#) and [IviDmm Read MultiPoint](#) VIs cause the DMM to leave the Idle state and transition to the Wait-For-Trigger state. The [IviDmm Read MultiPoint](#) VI does not return until the measurement process is complete and the DMM returns to the Idle state. The [IviDmm Initiate](#) VI returns as soon as the DMM leaves the Idle state.

The IviDmmMultiPoint extension group does not add additional capabilities to the Wait-For-Trigger state.

After the DMM leaves the Wait-For-Trigger state, it then executes a delay. The length of the delay is specified by the [IviDmm Trigger Delay](#) property. After the measurement is taken, the DMM then, if it is capable of doing so, generates the Measurement Complete signal.

The DMM then compares the sample count with the number of measurements taken since the last trigger event. The sample count is specified by the [IviDmm Sample Count \[MP\]](#) property. If the number of measurements is not equal to the sample count the DMM moves to the Wait-For-Sample-Trigger state. The DMM remains in the Wait-For-Sample-Trigger state until the event specified by the property [IviDmm Sample Trigger \[MP\]](#) occurs. Then it takes another measurement.

Once the number of measurements taken is equal to the sample count, the DMM then compares the number of trigger events that have occurred since either the [IviDmm Initiate](#) or [IviDmm Read MultiPoint](#) VI was called. The trigger count is specified by

the property [IviDmm Trigger Count \[MP\]](#). If the number of trigger events is not equal to the trigger count, the DMM returns to the Wait-For-Trigger state.

Once the number of trigger events is equal to the trigger count, the DMM returns to the Idle state. The [IviDmm Fetch MultiPoint](#) VI is used to retrieve measured data from measurements initiated by the [IviDmm Initiate](#) VI. The measurement data returned from the [IviDmm Read MultiPoint](#) and [IviDmm Fetch MultiPoint](#) VIs is acquired after the DMM has left the Wait-For-Trigger state.

IviDmmSoftwareTrigger Behavior Model

The behavior model of the IviDmmSoftwareTrigger follows the behavior model of the [IviDmmBase](#) capability group and the [IviDmmMultiPoint](#) extension group. It defines an additional trigger event for the trigger source.

The DMM leaves the Wait-For-Trigger state when it receives a trigger event specified by the [IviDmm Trigger Source](#) property. The DMM leaves the Wait-For-Sample-Trigger state when it receives a trigger event specified by the [IviDmm Sample Trigger \[MP\]](#) property. When the trigger source or sample trigger is set to [IVIDMM_VAL_SOFTWARE_TRIG](#), the [IviDmm Send Software Trigger](#) VI is used to generate the trigger event. Calling this VI causes the DMM to take a measurement.

IviDmm VI Tree

The VI tree for the IviDmm class driver, including IVI and VXI*plug&play* required VIs, is shown in the following table.



Note You use the LabVIEW Property Node to get and set properties.

IviDmm VI Tree

Name or Class	VI Name	Required By
Initialize	IviDmm Initialize	VPP
Initialize With Options	IviDmm Initialize With Options	IVI
Configuration		
Configure Measurement	IviDmm Configure Measurement	IviDmmConfigureMeasurement
Specific Measurements		
Configure AC Bandwidth [AC]	IviDmm Configure AC Bandwidth	IviDmmACMeasurement
Configure Frequency Voltage Range [FRQ]	IviDmm Configure Frequency Voltage Range	IviDmmFrequencyMeasurement
Temperature		
Configure Transducer Type [TMP]	IviDmm Configure Transducer Type	IviDmmTemperatureMeasurement
Configure Thermocouple [TC]	IviDmm Configure Thermocouple	IviDmmThermocouple
Configure Fixed Ref Junction [TC]	IviDmm Configure Fixed Ref Junction	IviDmmThermocouple
	IviDmm	IviDmmResistanceTemperatureDevice

Confident RTD [RTD] [Configure RTD](#)

Configure Thermistor [THM] [IviDmm Configure Thermistor](#) IviDmmThermistor

Trigger

Configure Trigger [IviDmm Configure Trigger](#) IviDmm

Configure Trigger Slope [IviDmm Configure Trigger Slope](#) IviDmmTriggerSlope

Multipoint

Configure Multipoint [MP] [IviDmm Configure Multipoint](#) IviDmmMultiPoint

Configure Meas Complete Dest [MP] [IviDmm Configure Meas Complete Dest](#) IviDmmMultiPoint

Measurement Operation Options

Configure Auto Zero Mode [AZ] [IviDmm Configure Auto Zero Mode](#) IviDmmAutoZero

Configure Power Line Frequency [PLF] [IviDmm Configure Power Line Frequency](#) IviDmmPowerLineFrequency

Configuration Information

Get Auto Range Value [ARV] [IviDmm Get Auto Range Value](#) IviDmmAutoRangeValue

Get Aperture Time Info [DI]	<u>IviDmm Get Aperture Time Info</u>	IviDmmDeviceInfo
Measurement		
Read	<u>IviDmm Read</u>	IviDmm
Read Multipoint [MP]	<u>IviDmm Read Multipoint</u>	IviDmmMultiPoint
Low-Level Measurement		
Initiate	<u>IviDmm Initiate</u>	IviDmm
Send Software Trigger	<u>IviDmm Send Software Trigger</u>	IviDmmSoftwareTrigger
Fetch	<u>IviDmm Fetch</u>	IviDmm
Fetch Multipoint [MP]	<u>IviDmm Fetch Multiple Point</u>	IviDmmMultiPoint
Abort	<u>IviDmm Abort</u>	IviDmm
Is Overrange	<u>IviDmm Is Over-Range</u>	IviDmm
Utility		
Reset	<u>IviDmm Reset</u>	VPP
Reset With Defaults	<u>Reset With Defaults</u>	VPP
Self-Test	<u>IviDmm Self-Test</u>	VPP
Disable	<u>Disable</u>	VPP
Invalidate All	<u>Invalidate All Attributes</u>	VPP

Attributes

Revision Query	IviDmm Revision Query	VPP
Error-Query	IviDmm Error-Query	VPP
Error Message	IviDmm Error Message	VPP

Interchangeability Info

Get Next Interchange Warning	IviDmm Get Next Interchange Warning	IVI
Reset Interchange Check	IviDmm Reset Interchange Check	IVI
Clear Interchange Warnings	IviDmm Clear Interchange Warnings	IVI

Coercion Info

Get Next Coercion Record	IviDmm Get Next Coercion Record	IVI
Close	IviDmm Close	VPP

Related Topics

[IviDmm Properties](#)

[IVI Inherent VIs](#)

IviDmm Properties



Note You use the LabVIEW Property Node to get and set properties.

IviDmm Base Properties

[IviDmm Function](#)

[IviDmm Range](#)

[IviDmm Resolution](#)

[IviDmm Trigger Source](#)

[IviDmm Trigger Delay](#)

ACMeasurement Properties

[IviDmm AC Minimum Frequency](#)

[IviDmm AC Maximum Frequency](#)

AutoRangeValue Properties

[IviDmm Auto Range Value](#)

AutoZero Properties

[IviDmm Auto Zero](#)

DeviceInfo Properties

[IviDmm Aperture Time \[DI\]](#)

[IviDmm Aperture Time Units \[DI\]](#)

FrequencyMeasurement Properties

[IviDmm Frequency Voltage Range](#)

Multipoint Properties

IviDmm Meas Complete Destination [MP]

IviDmm Sample Count [MP]

IviDmm Sample Interval [MP]

IviDmm Sample Trigger [MP]

IviDmm Trigger Count [MP]

PowerLineFrequency Property

[IviDmm Powerline Frequency](#)

ResistanceTemperatureDevice Properties

[IviDmm RTD Alpha](#)

[IviDmm RTD Resistance](#)

TemperatureMeasurement Properties

[IviDmm Transducer Type](#)

Thermistor Properties

[IviDmm Thermistor Resistance](#)

Thermocouple Properties

[IviDmm Thermocouple Type](#)

[IviDmm Reference Junction Type](#)

[IviDmm Fixed Reference Junction](#)

TriggerSlope Properties

[IviDmm Trigger Slope](#)

IviDmm Property Value Definitions

The following table defines values for each property constant in the IviDmm class.



Note In the following table, the literal string IVIDMM precedes the property name, and the literal string IVIDMM precedes the value extension name.

IviDmm Property Defined Values

Property	Value Name	Defined Value	Definitions
APERTURE TIME UNITS	Seconds	0	Reports that the units for the value returned by IviDmm Aperture Time seconds
	Power Line Cycles	1	Reports that the units for the value returned by IviDmm Aperture Time Powerline Cycles.
AUTO ZERO	Off	0	Disables auto-ranging. The DMM reports the range to the value it most recently calculated. Further queries of this property return the actual range.
	On	1	Configures the DMM to take a Zero Reading for each measurement. The DMM subtracts the Zero Reading from the value it measures.
	Once	2	Configures the DMM to calculate the range before the next measurement. The DMM uses this range value for all subsequent measurements. Further queries of this property should return the actual range.
FUNCTION	DC Volts	1	Sets the DMM to measure DC voltage
	AC Volts	2	Sets the DMM to measure AC voltage
	DC Current	3	Sets the DMM to measure DC current
	AC Current	4	Sets the DMM to measure AC current
	2 Wire Resistance	5	Sets the DMM to measure 2-wire resistance.
	4 Wire Resistance	101	Sets the DMM to measure 4-wire resistance.
	Frequency	104	Sets the DMM to measure frequency
	Period	105	Sets the DMM to measure period.

	AC + DC Volts	106	Sets the DMM to measure AC plus voltage.
	AC + DC Current	107	Sets the DMM to measure AC plus current.
	Temperature	108	Sets the DMM to measure temper in Celsius.
MEAS COMPLETE DEST	None	-1	The measurement complete signal routed.
	External	2	Routes the measurement-complet signal to the external connector.
	PXI TRIG0 or VXI TTL0	111	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL0.
	PXI TRIG1 or VXI TTL1	112	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL1.
	PXI TRIG2 or VXI TTL2	113	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL2.
	PXI TRIG3 or VXI TTL3	114	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL3.
	PXI TRIG4 or VXI TTL4	115	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL4.
	PXI TRIG5 or VXI TTL5	116	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL5.
	PXI TRIG6 or VXI TTL6	117	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL6.
	PXI TRIG7 or VXI TTL7	118	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL7.

	ECL0	119	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on ECL0.
	ECL1	120	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on ECL1.
	PXI Star	131	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on the PXI Sta trigger bus.
	RTSI0	140	
	RTSI1	141	
	RTSI2	142	
	RTSI3	143	
	RTSI4	144	
	RTSI5	145	
	RTSI6	146	
RANGE	On	-1.0	Configures the DMM to calculate t range before each measurement automatically. When this property to IVIDMM_VAL_AUTO_RANGE_ you can obtain the actual range th DMM is currently using by getting value of the IviDmm Auto Range \ property.
	Off	-2.0	Disables auto-ranging. The DMM : the range to the value it most rece calculated. Further queries of this property return the actual range.
	Once	-3.0	Configures the DMM to calculate t range before the next measureme DMM uses this range value for all subsequent measurements. Furth queries of this property should reti actual range.

SAMPLE TRIGGER	Immediate	1	1
	External	2	2
	Software Trigger Function	3	The DMM exits the Wait-For-Trigg state or the Wait-For-Sample-Trigg state when the IviDmm Send Soft Trigger VI executes.
	Interval	10	The DMM exits the Wait-For-Samp Trigger state when the length of tir specified by the IviDmm Sample Ir property elapses.
	PXI TRIG0 or VXI TTL0	111	111
	PXI TRIG1 or VXI TTL1	112	112
	PXI TRIG2 or VXI TTL2	113	113
	PXI TRIG3 or VXI TTL3	114	114
	PXI TRIG4 or VXI TTL4	115	115
	PXI TRIG5 or VXI TTL5	116	116
	PXI TRIG6 or VXI TTL6	117	117
	PXI TRIG7 or VXI TTL7	118	118
	ECL0	119	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on ECL0.
	ECL1	120	120
	PXI Star	131	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on the PXI Sta

			trigger bus.
	RTSI0	140	
	RTSI1	141	
	RTSI2	142	
	RTSI3	143	
	RTSI4	144	
	RTSI5	145	
	RTSI6	146	
TEMP TC REF JUNC TYPE	Internal	1	Sets the DMM to use a fixed value the thermocouple junction compensation.
	Fixed	2	Sets the DMM to use a fixed value the thermocouple junction compensation.
TEMP TC TYPE	B	1	Sets the DMM to measure temper from a B-type thermocouple.
	C	2	Sets the DMM to measure temper from a C-type thermocouple.
	D	3	Sets the DMM to measure temper from a D-type thermocouple.
	E	4	Sets the DMM to measure temper from a E-type thermocouple.
	G	5	Sets the DMM to measure temper from a G-type thermocouple.
	J	6	Sets the DMM to measure temper from a J-type thermocouple.
	K	7	Sets the DMM to measure temper from a K-type thermocouple.
	N	8	Sets the DMM to measure temper from a N-type thermocouple.
	R	9	Sets the DMM to measure temper from a R-type thermocouple.
	S	10	Sets the DMM to measure temper

			from a S-type thermocouple.
	T	11	Sets the DMM to measure temper from a T-type thermocouple.
	U	12	Sets the DMM to measure temper from a U-type thermocouple.
	V	13	Sets the DMM to measure temper from a V-type thermocouple.
TEMP TRANSDUCER TYPE	Thermocouple	1	Sets the DMM to measure temper using a thermocouple. Use the IviDmmThermocouple extension g to configure additional settings for transducer type.
	Thermistor	2	Sets the DMM to measure temper using a thermistor. Use the IviDmmThermistor extension grou configure additional settings for th transducer type.
	2-Wire RTD	3	Sets the DMM to measure temper using a 2-wire temperature resista device. Use the IviDmmResistanceTemperatureDe Extension Group to configure addi settings for this transducer type.
	4-Wire RTD	4	Sets the DMM to measure temper using a 4-wire temperature resista device. Use the IviDmmResistanceTemperatureDe Extension Group to configure addi settings for this transducer type.
TRIGGER DELAY	On	-1.0	Configures the DMM to calculate t trigger delay before each measure
	Off	-2.0	Stops the DMM from calculating th trigger delay. Sets the trigger dela; last trigger delay the DMM calcula  Note After you set this prope

			IVIDMM_VAL_AUTO_DELAY further queries of this property should return the actual delay
TRIGGER SLOPE	Positive	0	Sets the trigger event to occur on rising edge of the trigger pulse.
	Negative	1	Sets the trigger event to occur on falling edge of the trigger pulse.
TRIGGER SOURCE	Immediate	1	The DMM exits the Wait-For-Trigg state or the Wait-For-Sample-Trigg state immediately after entering. It not wait for a trigger of any kind.
	External	2	The DMM exits the Wait-For-Trigg state when a trigger occurs on the external trigger input.
	Software Trigger Function	3	The DMM exits the Wait-For-Trigg state or the Wait-For-Sample-Trigg state when the IviDmm Send Soft Trigger VI executes.
	PXI TRIG0 or VXI TTL0	111	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL0.
	PXI TRIG1 or VXI TTL1	112	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL1.
	PXI TRIG2 or VXI TTL2	113	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL2.
	PXI TRIG3 or VXI TTL3	114	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL3.
	PXI TRIG4 or VXI TTL4	115	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL4.
	PXI TRIG5 or VXI TTL5	116	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state

			it receives a trigger on TTL5.
	PXI TRIG6 or VXI TTL6	117	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL6.
	PXI TRIG7 or VXI TTL7	118	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on TTL7.
	ECLO	119	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on ECLO.
	ECL1	120	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on ECL1.
	PXI Star	131	The DMM exits the Wait-For-Trigg the Wait-For-Sample-Trigger state it receives a trigger on the PXI Sta trigger bus.
	RTSI0	140	
	RTSI1	141	
	RTSI2	142	
	RTSI3	143	
	RTSI4	144	
	RTSI5	145	
	RTSI6	146	

Related Topics

[IviDmm VI Tree](#)

[IviDmm Properties](#)

IviDmm Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviDmm Errors	0xBFFFA2001 to 0xBFFFA3FFF
IviDmm Warnings	0x3FFFA2001 to 0x3FFFA3FFF
IVI Specific Driver Errors	0xBFFFA4000 to 0xBFFFA5FFF
IVI Specific Driver Warnings	0x3FFFA4000 to 0x3FFFA5FFF
IVI Errors	0xBFFFA0000 to 0xBFFFA1FFF
IVI Warnings	0x3FFFA0000 to 0x3FFFA1FFF
Common Instrument Driver Errors	0xBFFFC0000 to 0xBFFFCFFFF
Common Instrument Driver Warnings	0x3FFFC0000 to 0x3FFFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0xFFFFFFF

The IviDmm class driver defines the error codes shown in the following table in addition to the IVI defined error codes.

IviDmm Error Codes

Error	Value	Message
IVIDMM_ERROR_MAX_TIME_EXCEEDED	0xBFFA2001	Maximum time exceeded before the operation completed.
IVIDMM_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001	The trigger source is not software trigger.

IviDmm Warning Code

Warning	Value	Message
IVIDMM_WARN_OVER_RANGE	0x3FFA2001	Over Range warning

Related Topic

[IVI Status Codes](#)

IviDmm VIs

Expand this book to view an alphabetized list of IviDmm VIs.

IviDmm Abort

IviDmm Base Capability Group

This VI aborts a previously initiated measurement and returns the DMM to the Idle state.



Note This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviDmm Error-Query](#) VI at the conclusion of the sequence.



instrument handle The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

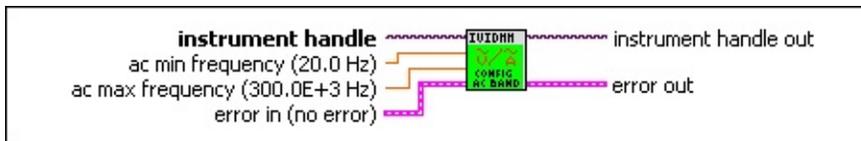
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure AC Bandwidth

IviDmmACMeasurement Capability Group [AC]

This VI configures the AC minimum and maximum frequency for DMMs that take AC voltage or AC current measurements.

This VI affects the behavior of the instrument only if the [IviDmm Function](#) property is set to an AC voltage or AC current measurement.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

ac min frequency (20.0 Hz) Pass the minimum expected frequency component of the input signal in Hertz. The driver sets the [IviDmm AC Minimum Frequency \[AC\]](#) property to this value.

Default Value: 20.0

ac max frequency (300.0E+3 Hz) Pass the maximum expected frequency component of the input signal in Hertz. The driver sets the [IviDmm AC Maximum Frequency \[AC\]](#) property to this value.

Default Value: 300000.0

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

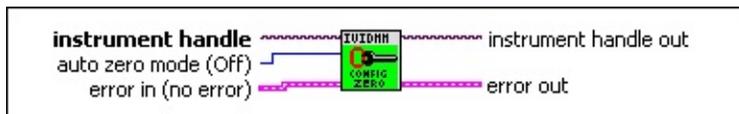
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Auto Zero Mode

IviDmmAutoZero Capability Group [AZ]

This VI configures the auto-zero mode of the DMM.



- instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.
- auto zero mode (Off)** Specify the auto-zero mode. The driver sets the [IviDmm Auto Zero \[AZ\]](#) property to this value.

When the auto-zero mode is enabled, the DMM internally disconnects the input signal and takes a Zero Reading. The DMM then subtracts the Zero Reading from the measurement. This prevents offset voltages present in the instrument's input circuitry from affecting measurement accuracy.

Defined Values:

IVIDMM_VAL_AUTO_ZERO_ON—Configures the DMM to take a Zero Reading for each measurement. The DMM subtracts the Zero Reading from the value it measures.

IVIDMM_VAL_AUTO_ZERO_OFF—Disables the auto-zero feature.

IVIDMM_VAL_AUTO_ZERO_ONCE—Configures the DMM to take a Zero Reading immediately. The DMM then subtracts this Zero Reading from all subsequent values it measures.

Default Value: IVIDMM_VAL_AUTO_ZERO_OFF

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

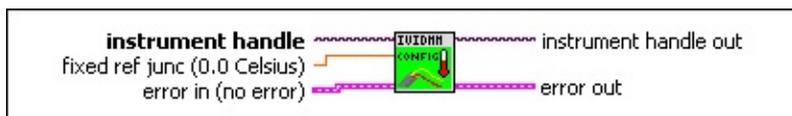
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Fixed Ref Junction

IviDmmThermocouple Capability Group [TC]

This VI configures the fixed reference junction for a thermocouple with a fixed reference junction type.

This VI affects the behavior of the instrument only when the IviDmm [Reference Junction Type \[TC\]](#) property is set to IVIDMM_VAL_TEMP_REF_JUNC_FIXED.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

fixed ref junc (0.0° Celsius) Pass the reference junction temperature when a fixed reference junction type thermocouple is used to take the temperature measurement. The units are degrees Celsius. The driver sets the [IviDmm Fixed Reference Junction \[TC\]](#) property to this value.

Default Value: 0.0 °C

 **Note** This property may also be used to specify the thermocouple junction temperature of an instrument that does not have an internal temperature sensor.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information

out of a VI to be used by other VIs.

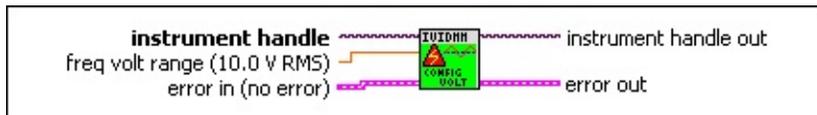
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Frequency Voltage Range

IviDmmFrequencyMeasurement Capability Group [FRQ]

This VI configures the frequency voltage range of the DMM for frequency and period measurements.

This VI affects the behavior of the instrument only if the [IviDmm Function](#) property is set to IVIDMM_VAL_FREQ or IVIDMM_VAL_PERIOD.



I/O **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

freq volt range (10.0 V RMS) Pass the expected maximum value of the input signal for frequency and period measurements in volts RMS. The driver sets the [IviDmm Frequency Voltage Range \[FRQ\]](#) property to this value.

The driver reserves special negative values for the auto-range mode.

Defined Values:

IVIDMM_VAL_AUTO_RANGE_ON (-1.0)—Auto-range On

IVIDMM_VAL_AUTO_RANGE_OFF (-2.0)—Auto-range Off

Default Value: 10.0 V RMS

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



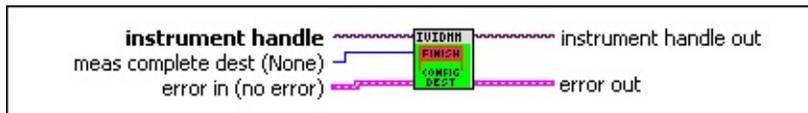
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Meas Complete Dest

IviDmmMultiPoint Capability Group [MP]

This VI configures the destination of the measurement-complete signal. This signal is commonly referred to as Voltmeter Complete.



- instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.
- meas complete dest (None)** Specify the destination of the measurement-complete signal. The driver sets the [IviDmm Meas Complete Destination \[MP\]](#) property to this value.

Defined Values:

IVIDMM_VAL_NONE—None

IVIDMM_VAL_EXTERNAL—External

IVIDMM_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVIDMM_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVIDMM_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVIDMM_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVIDMM_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVIDMM_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVIDMM_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVIDMM_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVIDMM_VAL_ECL0—VXI ECL0

IVIDMM_VAL_ECL1—VXI ECL1

IVIDMM_VAL_PXI_STAR—PXI Star

IVIDMM_VAL_RTSL_0—RTSL line 0

IVIDMM_VAL_RTISI_1—RTSI line 1
IVIDMM_VAL_RTISI_2—RTSI line 2
IVIDMM_VAL_RTISI_3—RTSI line 3
IVIDMM_VAL_RTISI_4—RTSI line 4
IVIDMM_VAL_RTISI_5—RTSI line 5
IVIDMM_VAL_RTISI_6—RTSI line 6
Default Value: IVIDMM_VAL_NONE



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



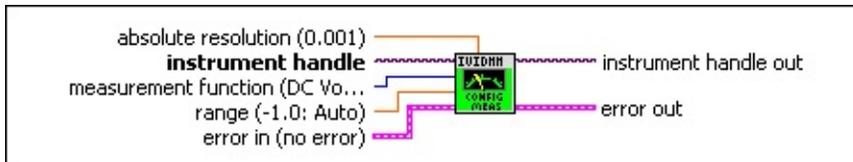
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Measurement

IviDmm Base Capability Group

This VI configures the common properties of the DMM. These properties include the measurement function, maximum range, and resolution.



instrument handle The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

measurement function (DC Volts) Pass the measurement VI you want the DMM to perform. The driver sets the [IviDmm Function](#) property to this value.

Valid Values:

IVIDMM_VAL_DC_VOLTS—DC Volts

IVIDMM_VAL_AC_VOLTS—AC Volts

IVIDMM_VAL_DC_CURRENT—DC Current

IVIDMM_VAL_AC_CURRENT—AC Current

IVIDMM_VAL_2_WIRE_RES—2-Wire Resistance

IVIDMM_VAL_4_WIRE_RES—4-Wire Resistance

IVIDMM_VAL_AC_PLUS_DC_VOLTS—AC plus DC Volts

IVIDMM_VAL_AC_PLUS_DC_CURRENT—AC plus DC Current

IVIDMM_VAL_FREQ—Frequency

IVIDMM_VAL_PERIOD—Period

IVIDMM_VAL_TEMPERATURE—Temperature (C)

Default Value:

IVIDMM_VAL_DC_VOLTS

range (-1.0: Auto) Pass the measurement range you want to use. The driver sets the [IviDmm Range](#) property to this value.

Use positive values to represent the absolute value of the maximum expected measurement. The value must be in units appropriate for the Measurement Function. For example, when you set the Measurement Function to `IVIDMM_VAL_DC_VOLTS`, you must specify the Range in volts. Setting this parameter to 10.0 configures the DMM to measure DC voltages from -10.0 to +10.0 volts.

The driver reserves special negative values for controlling the DMM's auto-ranging capability.

Defined Values:

`IVIDMM_VAL_AUTO_RANGE_ON` (-1.0)—Auto-range On

`IVIDMM_VAL_AUTO_RANGE_OFF` (-2.0)—Auto-range Off

`IVIDMM_VAL_AUTO_RANGE_ONCE` (-3.0)—Auto-range Once

Default Value:

`IVIDMM_VAL_AUTO_RANGE_ON` (-1.0)



Notes

1. Setting this parameter to `IVIDMM_VAL_AUTO_RANGE_ONCE` configures the DMM to auto-range once, turn auto-range off, and then remain at the current maximum range.
2. Setting this to parameter `IVIDMM_VAL_AUTO_RANGE_OFF` configures the DMM to stop auto-ranging and keep the range fixed at the current maximum range.



absolute resolution (0.001) Pass your desired measurement resolution in absolute units. The driver sets the [IviDmm Resolution](#) property to this value.

The value must be in units appropriate for the Measurement function.

Setting this parameter to lower values increases the measurement accuracy. Setting this parameter to higher values increases the measurement speed.

Default Value: 0.001



Note This parameter is ignored if the **Range** parameter is set to `IVIDMM_VAL_AUTO_RANGE_ON`.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



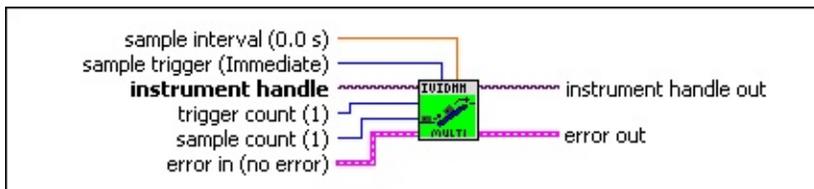
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure MultiPoint

IviDmmMultiPoint Capability Group [MP]

This VI configures the properties that affect multipoint measurements. These properties are the trigger count, sample count, sample trigger and sample interval.



instrument handle The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

trigger count (1) Pass the number of triggers you want the DMM to receive before returning to the Idle state. The driver sets the [IviDmm Trigger Count \[MP\]](#) property to this value.

Default Value: 1

sample count (1) Pass the number of measurements you want the DMM to take each time it receives a trigger. The driver sets the [IviDmm Sample Count \[MP\]](#) property to this value.

Default Value: 1

sample trigger (Immediate) Pass the type of sample trigger you want to use. The driver sets the [IviDmm Sample Trigger \[MP\]](#) property to this value.

When the DMM takes a measurement and the **Sample Count** parameter is greater than 1, the DMM does not take the next measurement until the event you specify in the **Sample Trigger** parameter occurs.

Valid Values:

IVIDMM_VAL_IMMEDIATE—Immediate

IVIDMM_VAL_EXTERNAL—External

IVIDMM_VAL_SOFTWARE_TRIG—Software Trigger Function

IVIDMM_VAL_INTERVAL—Interval
IVIDMM_VAL_TTL0—PXI TRIG0 or VXI TTL0
IVIDMM_VAL_TTL1—PXI TRIG1 or VXI TTL1
IVIDMM_VAL_TTL2—PXI TRIG2 or VXI TTL2
IVIDMM_VAL_TTL3—PXI TRIG3 or VXI TTL3
IVIDMM_VAL_TTL4—PXI TRIG4 or VXI TTL4
IVIDMM_VAL_TTL5—PXI TRIG5 or VXI TTL5
IVIDMM_VAL_TTL6—PXI TRIG6 or VXI TTL6
IVIDMM_VAL_TTL7—PXI TRIG7 or VXI TTL7
IVIDMM_VAL_ECL0—VXI ECL0
IVIDMM_VAL_ECL1—VXI ECL1
IVIDMM_VAL_PXI_STAR—PXI Star
IVIDMM_VAL_RTSL_0—RTSL line 0
IVIDMM_VAL_RTSL_1—RTSL line 1
IVIDMM_VAL_RTSL_2—RTSL line 2
IVIDMM_VAL_RTSL_3—RTSL line 3
IVIDMM_VAL_RTSL_4—RTSL line 4
IVIDMM_VAL_RTSL_5—RTSL line 5
IVIDMM_VAL_RTSL_6—RTSL line 6
Default Value: IVIDMM_VAL_IMMEDIATE



Notes

1. IVIDMM_VAL_IMMEDIATE—The DMM takes the next measurement immediately. The DMM does not wait for a trigger of any kind.
2. IVIDMM_VAL_EXTERNAL—The DMM takes the next measurement when a trigger occurs on the external trigger input.
3. IVIDMM_VAL_SOFTWARE_TRIG—The DMM takes the next measurement when you call the Send Software Trigger [SWT] VI.
4. IVIDMM_VAL_INTERVAL—The DMM takes the next

measurement after waiting the length of time you specify in the **Sample Interval** parameter.

This parameter is ignored if the value of the **Sample Count** parameter is 1.

- sample interval (0.0 s)** Pass the length of time you want the DMM to wait between samples. Express this value in seconds. The driver sets the [IviDmm Sample Interval \[MP\]](#) property to this value.

If the **Sample Count** parameter is greater than 1 and the **Sample Trigger** parameter is set to `IVIDMM_VAL_INTERVAL`, the DMM waits between measurements for the length of time you specify with this parameter. Otherwise, this property does not affect the behavior of the instrument.

Default Value: 0.0 seconds



Notes

1. This parameter is ignored if the value of the **Sample Count** parameter is 1.
2. This parameter is ignored if the value of the **Sample Trigger** parameter is not `IVIDMM_VAL_INTERVAL`.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



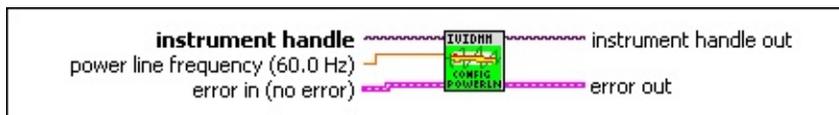
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Power Line Frequency

IviDmmPowerLineFrequency Capability Group [PLF]

This VI configures the powerline frequency of the DMM.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

powerline frequency (60.0 Hz) Specify the powerline frequency in Hertz. The driver sets the [IviDmm Powerline Frequency \[PLF\]](#) property to this value.

Default Value: 60.0 Hz

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

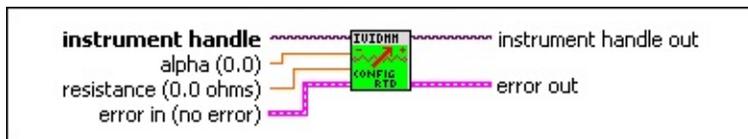
IviDmm Configure RTD

IviDmmResistanceTemperatureDevice Capability Group [RTD]

This VI configures the **alpha** and **resistance** parameters for a resistance temperature device.

This VI affects the behavior of the instrument only when the [IviDmm Transducer Type \[TMP\]](#) property is set to `IVIDMM_VAL_2_WIRE_RTD` or `IVIDMM_VAL_4_WIRE_RTD`.

 **Note** The driver assumes that you are using a Platinum Resistance Temperature Device.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

alpha (0.0) Pass the **alpha** parameter for a resistance temperature device. The driver sets the [IviDmm RTD Alpha \[RTD\]](#) property to this value.

Default Value: 0.0

resistance (0.0 ohms) Specifies the **R0** parameter (resistance) for a resistance temperature device. The RTD resistance is also known as the RTD reference value. The driver sets the [IviDmm RTD Resistance \[RTD\]](#) property to this value.

Default Value: 0.0 ohms

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

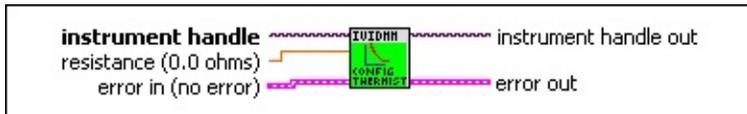
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Thermistor

IviDmmThermistor Capability Group [THM]

This VI configures the resistance for a thermistor temperature measurement device.

This VI affects the behavior of the instrument only when the [IviDmm Transducer Type](#) property is set to IVIDMM_VAL_THERMISTOR.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

resistance (0.0 ohms) Pass the resistance of the thermistor in Ohms. The driver sets the [IviDmm Thermistor Resistance \[THM\]](#) property to this value.

Default Value: 0.0 ohms

 **Note** The driver assumes that you are using an interchangeable thermistor. Interchangeable thermistors are thermistors that exhibit similar behavior for a given resistance value.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

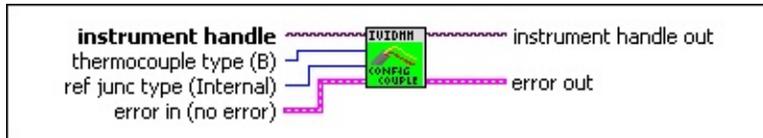
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Thermocouple

IviDmmThermocouple Capability Group [TC]

This VI configures the thermocouple type and the reference junction type of the thermocouple for DMMs that take temperature measurements using a thermocouple transducer type.

This VI affects the behavior of the instrument only if the [IviDmm Transducer Type \[TMP\]](#) property is set to `IVIDMM_VAL_THERMOCOUPLE`.



- instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.
- thermocouple type (B)** Pass the type of thermocouple used to measure the temperature. The driver uses this value to set the [IviDmm Thermocouple Type \[TC\]](#) property.

Defined Values:

IVIDMM_VAL_TEMP_TC_B
IVIDMM_VAL_TEMP_TC_C
IVIDMM_VAL_TEMP_TC_D
IVIDMM_VAL_TEMP_TC_E
IVIDMM_VAL_TEMP_TC_J
IVIDMM_VAL_TEMP_TC_K
IVIDMM_VAL_TEMP_TC_N
IVIDMM_VAL_TEMP_TC_R
IVIDMM_VAL_TEMP_TC_S
IVIDMM_VAL_TEMP_TC_T
IVIDMM_VAL_TEMP_TC_U
IVIDMM_VAL_TEMP_TC_V

Default Value: `IVIDMM_VAL_TEMP_TC_B`

- ref junc type (Internal)** Pass the type of reference junction to be used in the reference junction compensation of a thermocouple measurement. The driver uses this value to set the [IviDmm Reference Junction Type \[TC\]](#) property.

Defined Values:

`IVIDMM_VAL_TEMP_REF_JUNC_INTERNAL`—The DMM uses an internal reference junction type.

`IVIDMM_VAL_TEMP_REF_JUNC_FIXED`—The DMM uses a fixed value for the reference junction. Use the [IviDmm Configure Fixed Ref Junction \[TC\]](#) VI to specify the fixed reference junction value.

Default Value: `IVIDMM_VAL_TEMP_REF_JUNC_INTERNAL`

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

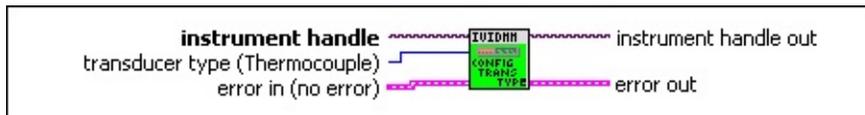
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Transducer Type

IviDmmTemperatureMeasurement Capability Group [TMP]

This VI configures the DMM to take temperature measurements from a specified transducer type.

This VI affects the behavior of the instrument only when the [IviDmm Function](#) property is set to .



I/O **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

transducer type (Thermocouple) Pass the type of device used to measure the temperature. The driver uses this value to set the [IviDmm Transducer Type \[TMP\]](#) property.

Defined Values:

IVIDMM_VAL_THERMOCOUPLE—Thermocouple

IVIDMM_VAL_THERMISTOR—Thermistor

IVIDMM_VAL_2_WIRE_RTD—2 Wire Resistance Temperature Device

IVIDMM_VAL_4_WIRE_RTD—4 Wire Resistance Temperature Device

Default Value:

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



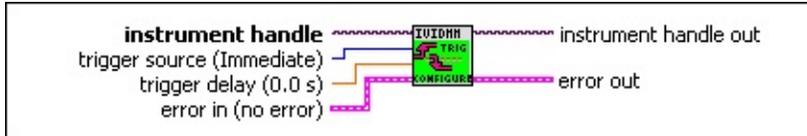
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Trigger

IviDmm Base Capability Group

This VI configures the common DMM trigger properties. These properties are the trigger source and trigger delay.



- instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.
- trigger source (Immediate)** Specify the trigger source you want to use. The driver sets the [IviDmm Trigger Source](#) property to this value.

After you call the Read or Initiate VI, the DMM waits for the trigger you specify in this parameter. After it receives the trigger, the DMM waits the length of time you specify in the **Trigger Delay** parameter. The DMM then takes a measurement.

Valid Values:

- IVIDMM_VAL_IMMEDIATE—Immediate
- IVIDMM_VAL_EXTERNAL—External
- IVIDMM_VAL_SOFTWARE_TRIG—Software Trigger Function
- IVIDMM_VAL_TTL0—PXI TRIG0 or VXI TTL0
- IVIDMM_VAL_TTL1—PXI TRIG1 or VXI TTL1
- IVIDMM_VAL_TTL2—PXI TRIG2 or VXI TTL2
- IVIDMM_VAL_TTL3—PXI TRIG3 or VXI TTL3
- IVIDMM_VAL_TTL4—PXI TRIG4 or VXI TTL4
- IVIDMM_VAL_TTL5—PXI TRIG5 or VXI TTL5
- IVIDMM_VAL_TTL6—PXI TRIG6 or VXI TTL6
- IVIDMM_VAL_TTL7—PXI TRIG7 or VXI TTL7
- IVIDMM_VAL_ECL0—VXI ECL0

IVIDMM_VAL_ECL—VXI ECL1
IVIDMM_VAL_PXI_STAR—PXI Star
IVIDMM_VAL_RTSL_0—RTSL line 0
IVIDMM_VAL_RTSL_1—RTSL line 1
IVIDMM_VAL_RTSL_2—RTSL line 2
IVIDMM_VAL_RTSL_3—RTSL line 3
IVIDMM_VAL_RTSL_4—RTSL line 4
IVIDMM_VAL_RTSL_5—RTSL line 5
IVIDMM_VAL_RTSL_6—RTSL line 6
Default Value: IVIDMM_VAL_IMMEDIATE



Notes

1. IVIDMM_VAL_IMMEDIATE—The DMM does not wait for a trigger of any kind.
2. IVIDMM_VAL_EXTERNAL—The DMM waits for a trigger on the external trigger input.
3. IVIDMM_VAL_SOFTWARE_TRIG—The DMM waits until you call the Send Software Trigger [SWT] VI.



trigger delay (0.0 s) Pass the value you want to use for the trigger delay. Express this value in seconds. The driver sets the [IviDmm Trigger Delay](#) property to this value.

The trigger delay specifies the length of time the DMM waits after it receives the trigger and before it takes a measurement.

Use positive values to set the trigger delay in seconds. The driver reserves negative values for configuring the DMM to calculate the trigger delay automatically.

Default Value: 0.0 seconds



Notes

1. Setting this parameter to IVIDMM_VAL_AUTO_DELAY_ON (-1.0) configures the DMM to calculate the trigger delay before each measurement.
2. Setting this parameter to IVIDMM_VAL_AUTO_DELAY_OFF

(-2.0) stops the DMM from calculating the trigger delay and sets the trigger delay to the last automatically calculated value.

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

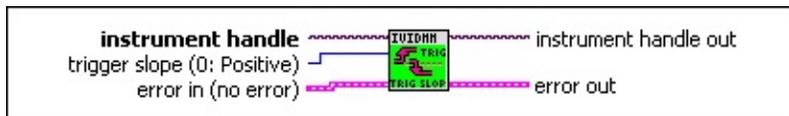
 **[error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Configure Trigger Slope

IviDmmTriggerSource Capability Group [TS]

This VI configures the polarity of the external trigger source of the DMM.



I/O **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

trigger slope (0: Positive) Pass the polarity of the external trigger slope. The driver sets the [IviDmm Trigger Slope \[TS\]](#) property to this value. The DMM triggers on either the rising or the falling edge of the external trigger source depending on the value of this property.

Defined Values:

IVIDMM_VAL_POSITIVE - The driver triggers on the rising edge of the external trigger.

IVIDMM_VAL_NEGATIVE - The driver triggers on the falling edge of the external trigger.

Default: IVIDMM_VAL_POSITIVE

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

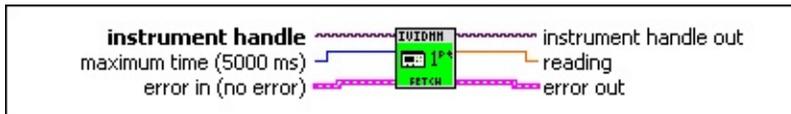
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Fetch

IviDmm Base Capability Group

This VI returns the value from a previously initiated measurement.

You must first call the [IviDmm Initiate](#) VI to initiate a measurement before calling this VI.



Notes

1. After this VI executes, the **Reading** parameter contains an actual reading or a value indicating that an overrange condition occurred.
2. If an overrange condition occurs, the **Reading** parameter contains an IEEE-defined NaN (Not a Number) value and the VI returns the IVIDMM_WARN_OVER_RANGE (0x3FFA2001) warning code.
3. You can test the measurement value for an overrange condition by calling the IviDmm Is Over-Range VI.
4. The class driver returns a simulated measurement when this VI is called and the [IviDmm Simulate](#) property is set to TRUE and the [IviDmm Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to **National Instruments IVI Driver Help»IVI Class Driver Simulation Overview**.
5. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviDmm Error-Query](#) VI at the conclusion of the sequence.



instrument handle The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

- maximum time (5000 ms)** Pass the maximum length of time in which to allow the fetch operation to complete. Express this value in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIDMM_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2003) error code. When this occurs, you can call `IviDmm Abort` to cancel the fetch operation and return the instrument to the Idle state.

Defined Values:

`IVIDMM_VAL_MAX_TIME_INFINITE (-1)`—Wait indefinitely for a timeout.

`IVIDMM_VAL_MAX_TIME_IMMEDIATE (0)`—Do not wait for a timeout.

Default Value: 5000 (ms)



Note The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

- reading** Returns the measured value. The value you specify for the **Measurement Function** parameter of the [IviDmm Configure Measurement](#) VI determines the units of this parameter as shown in the following table.

DC Volts	volts
AC Volts	volts
DC Current	amperes
AC Current	amperes
2-Wire Resistance	ohms
4-Wire Resistance	ohms

AC plus DC Volts	volts
AC plus DC Current	amperes
Temperature	Celsius
Frequency	hertz
Period	seconds



Notes

1. If an overrange condition occurs, the **Reading** parameter contains an IEEE-defined NaN (Not a Number) value and the VI returns the `IVIDMM_WARN_OVER_RANGE (0x3FFA2001)` warning code.
2. You can test the measurement value for an overrange condition by calling the [IviDmm Is Over-Range](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

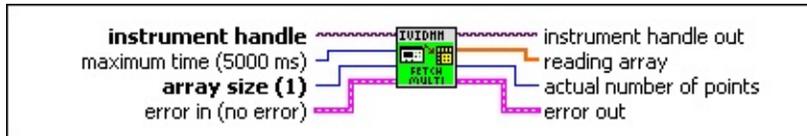
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Fetch MultiPoint

IviDmmMultiPoint Capability Group [MP]

This VI returns an array of values from a previously initiated multipoint measurement. The number of measurements the DMM takes is determined by the values you specify for the **Trigger Count** and **Sample Count** parameters of the [IviDmm Configure Multi-Point \[MP\]](#) VI.

Call the IviDmm Initiate VI to initiate a measurement before calling this VI.



Notes

1. After this VI executes, each element in the **Reading Array** parameter is an actual reading or a value indicating that an overrange condition occurred.
2. If an overrange condition occurs, the corresponding Reading Array element contains an IEEE-defined NaN (Not a Number) value and the VI returns IVIDMM_WARN_OVER_RANGE (0x3FFA2001).
3. You can test each element in the **Reading Array** parameter for overrange with the IviDmm Is Over-Range VI.
4. The class driver returns a simulated measurement when this VI is called and the [IviDmm Simulate](#) property is set to TRUE and the [IviDmm Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to **National Instruments IVI Driver Help»IVI Class Driver Simulation Overview**.
5. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviDmm Error-Query](#) VI at the conclusion of the sequence.

 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

maximum time (5000 ms) Pass the maximum length of time in which to allow the multipoint fetch operation to complete expressed in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIDMM_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2003) error code. When this occurs, you can call [IviDmm Abort](#) to cancel the multipoint fetch operation and return the instrument to the Idle state.

Defined Values:

`IVIDMM_VAL_MAX_TIME_INFINITE` (1)—Wait indefinitely for a timeout.

`IVIDMM_VAL_MAX_TIME_IMMEDIATE` (0)—Do not wait for a timeout.

Default Value: 5000 ms

 **Note** The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.

array size (1) Pass the number of elements in the **Reading Array** parameter.

Default Value: 1

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

reading array Returns an array of the most recent measurement values. The value you specify for the **Measurement Function** parameter of the [IviDmm Configure Measurement](#) VI determines the units of this parameter, as shown in the following table.

DC Volts	volts
AC Volts	volts
DC Current	amperes
AC Current	amperes
2-Wire Resistance	ohms
4-Wire Resistance	ohms
AC plus DC Volts	volts
AC plus DC Current	amperes
Temperature	Celsius
Frequency	hertz
Period	seconds



Notes

1. If an overrange condition occurs, the corresponding Reading Array element contains an IEEE-defined NaN (Not a Number) value and the VI returns the IVIDMM_WARN_OVER_RANGE (0x3FFA2001) warning code.
2. You can test each element in the **Reading Array** parameter for an overrange condition by calling the [IviDmm Is Over-Range](#) VI.



actual number of points Indicates the number of measured values the VI places in the **Reading Array** parameter.



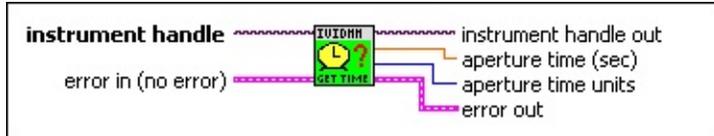
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Get Aperture Time Info

IviDmmDeviceInfo Capability Group [DI]

This VI returns additional information about the state of the instrument. Specifically, it returns the aperture time and the aperture time units.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

aperture time (sec) Returns the value of the [IviDmm Aperture Time \[DI\]](#) property.

The units of this property depend on the value of the [IviDmm Aperture Time Units \[DI\]](#) property.

aperture time units Returns the value of the [IviDmm Aperture Time Units \[DI\]](#) property.

Defined Return Values:

IVIDMM_VAL_SECONDS (0)—Seconds

IVIDMM_VAL_POWER_LINE_CYCLES (1)—Powerline cycles

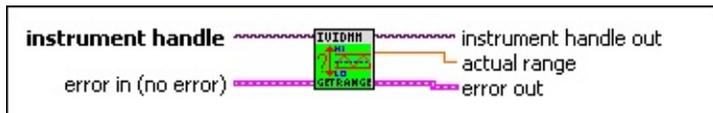
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Get Auto Range Value

IviDmmAutoRangeValue Capability Group [ARV]

This VI returns the actual range the DMM is currently using, even while it is auto-ranging.



I/O **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

actual range Returns the value of the [IviDmm Auto Range Value](#) property.

The units of the returned value depend on the value of the [IviDmm Function](#) property.

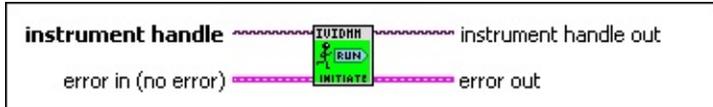
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Initiate

IviDmm Base Capability Group

This VI initiates a measurement. After you call this VI, the DMM leaves the Idle state and waits for a trigger. [Details](#).



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Initiate Details

This VI applies default values to properties that have not been set by you under the following conditions:

1. If you have not set the value of any property in the IviDmmMultiPoint extension group, the VI sets the [IviDmm Sample Count \[MP\]](#) and [IviDmm Trigger Count \[MP\]](#) properties to 1.
2. If you have not set the value of the [IviDmm Auto Zero \[AZ\]](#) property, the VI sets the value of this property to

IVIDMM_VAL_AUTO_ZERO_OFF (0).



Notes

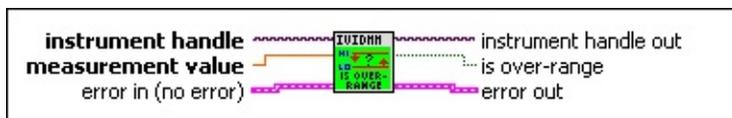
1. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviDmm Error-Query](#) VI at the conclusion of the sequence.
2. This VI performs interchangeability checking when the [IviDmm Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviDmm Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviDmm Interchange Check](#) property.
3. The class driver initiates a simulated measurement when this VI is called and the [IviDmm Simulate](#) property is set to TRUE and the [IviDmm Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to **National Instruments IVI Driver Help»IVI Class Driver Simulation Overview**.

IviDmm Is Over-Range

IviDmm Base Capability Group

This VI takes a measurement value that you obtain from either the Read or Fetch VIs and determines if the value is a valid measurement value or a value indicating that an overrange condition occurred.

 **Note** This VI does not check the instrument status.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

measurement value Pass the measurement value that you obtain from either the Read or Fetch VI. The driver tests this value to determine if the value is a valid measurement value or a value indicating that an overrange condition occurred.

Default Value: 0.0

 **Note** If an overrange condition occurs, the measurement value contains an IEEE-defined NaN (Not a Number) value indicating that an overrange occurred.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

is overrange Returns whether the measured value is a valid measurement or a value indicating that an overrange condition occurred.

Return Values:

True—The value indicates an overrange condition occurred.

False—The value is a valid measurement.



Note If an overrange condition occurs, the measurement value contains an IEEE-defined NaN (Not a Number) value indicating that an overrange occurred.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

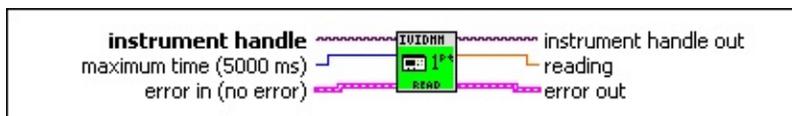
IviDmm Read

IviDmm Base Capability Group

This VI initiates a measurement, waits until the DMM has returned to the Idle state, and returns the measured value.

This VI applies default values to properties that have not been set by you under the following conditions:

1. If you have not set the value of any property in the IviDmmMultiPoint extension group, the VI sets the [IviDmm Sample Count \[MP\]](#) and [IviDmm Trigger Count \[MP\]](#) properties to 1.
2. If you have not set the value of the [IviDmm Auto Zero \[AZ\]](#) property, the VI sets the value of this property to IVIDMM_VAL_AUTO_ZERO_OFF (0).



Notes

1. After this VI executes, the **Reading** parameter contains an actual reading or a value indicating that an overrange condition occurred.
2. If an overrange condition occurs, the **Reading** parameter contains an IEEE-defined NaN (Not a Number) value and the VI returns the IVIDMM_WARN_OVER_RANGE (0x3ffA2001) warning code.
3. You can test the measurement value for an overrange condition by calling the [IviDmm Is Over-Range](#) VI.
4. This VI performs interchangeability checking when the [IviDmm Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviDmm Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviDmm Interchange](#)

[Check](#) property.

5. The class driver returns a simulated measurement when this VI is called and the [IviDmm Simulate](#) property is set to TRUE and the [IviDmm Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to **National Instruments IVI Driver Help»IVI Class Driver Simulation Overview**.



instrument handle The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



maximum time (5000 ms) Pass the maximum length of time in which to allow the read operation to complete. Express this value in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIDMM_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2003) error code. When this occurs, you can call `IviDmm Abort` to cancel the read operation and return the instrument to the Idle state.

Defined Values:

`IVIDMM_VAL_MAX_TIME_INFINITE` (-1)—Wait indefinitely for a timeout.

`IVIDMM_VAL_MAX_TIME_IMMEDIATE` (0)—Do not wait for a timeout.

Default Value: 5000 (ms)



Note The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



reading Returns the measured value. The value you specify for the

Measurement Function parameter of the [IviDmm Configure Measurement](#) VI determines the units of this parameter as shown in the following table.

DC Volts	volts
AC Volts	volts
DC Current	amperes
AC Current	amperes
2-Wire Resistance	ohms
4-Wire Resistance	ohms
AC plus DC Volts	volts
AC plus DC Current	amperes
Temperature	Celsius
Frequency	hertz
Period	seconds



Notes

1. If an overrange condition occurs, the **Reading** parameter contains an IEEE-defined NaN (Not a Number) value and the VI returns the `IVIDMM_WARN_OVER_RANGE (0x3FFA2001)` warning code.
2. You can test the measurement value for an overrange condition by calling the `IviDmm Is Over-Range VI`.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

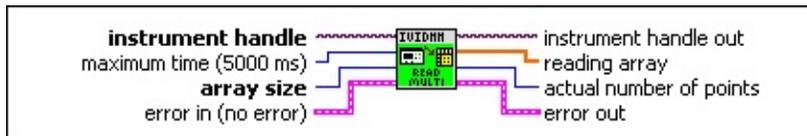
IviDmm Read MultiPoint

IviDmmMultiPoint Capability Group [MP]

This VI initiates the measurement, waits for the DMM to return to the Idle state, and returns an array of measured values. The number of measurements the DMM takes is determined by the values you specify for the **Trigger Count** and **Sample Count** parameters of the [IviDmm Configure MultiPoint \[MP\]](#) VI.

This VI applies default values to properties that you have not set under the following conditions:

1. If you have not set the value of any property in the IviDmmMultiPoint extension group, the VI sets the [IviDmm Sample Count \[MP\]](#) and [IviDmm Trigger Count \[MP\]](#) properties to 1.
2. If you have not set the value of the [IviDmm Auto Zero \[AZ\]](#) property, the VI sets the value of this property to `IVIDMM_VAL_AUTO_ZERO_OFF (0)`.



Notes

1. After this VI executes, each element in the **Reading Array** parameter is an actual reading or a value indicating that an overrange condition occurred.
2. If an overrange condition occurs, the corresponding Reading Array element contains an IEEE-defined NaN (Not a Number) value and the VI returns the `IVIDMM_WARN_OVER_RANGE (0x3FFA2001)` warning code.
3. You can test each element in the **Reading Array** parameter for an overrange condition by calling the [IviDmm Is Over-Range](#) VI.
4. This VI performs interchangeability checking when the [IviDmm](#)

[Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviDmm Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviDmm Interchange Check](#) property.

5. The class driver initiates an array of simulated measurements when this VI is called and the [IviDmm Simulate](#) property is set to TRUE and the [IviDmm Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to **National Instruments IVI Driver Help»IVI Class Driver Simulation Overview**.

 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

maximum time (5000 ms) Pass the maximum length of time in which to allow the multipoint read operation to complete. Express this value in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIDMM_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2003) error code. When this occurs, you can call the [IviDmm Abort](#) VI to cancel the multipoint read operation and return the instrument to the Idle state.

Defined Values:

`IVIDMM_VAL_MAX_TIME_INFINITE (-1)`—Wait indefinitely for a timeout.

`IVIDMM_VAL_MAX_TIME_IMMEDIATE (0)`—Do not wait for a timeout.

Default Value: 5000 ms

 **Note** The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.

array size Pass the number of elements in the **Reading Array** parameter.

Default Value: None

 **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.



reading array Returns an array of the most recent measurement values. The value you specify for the **Measurement Function** parameter of the [IviDmm Configure Measurement](#) VI determines the units of this parameter as shown in the following table:

DC Volts	volts
AC Volts	volts
DC Current	amperes
AC Current	amperes
2-Wire Resistance	ohms
4-Wire Resistance	ohms
AC plus DC Volts	volts
AC plus DC Current	amperes
Temperature	Celsius
Frequency	hertz
Period	seconds



Notes

1. If an overrange condition occurs, the corresponding Reading Array element contains an IEEE-defined NaN (Not a Number) value and the VI returns the `IVIDMM_WARN_OVER_RANGE` (0x3FFA2001) warning code.
2. You can test each element in the **Reading Array** parameter for an overrange condition by calling the [IviDmm Is Over-Range](#) VI.



actual number of points Indicates the number of measured values the VI places in the **Reading Array** parameter.



error out The error out cluster passes error or warning information

out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

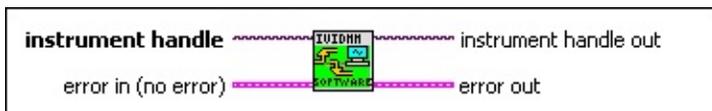
IviDmm Send Software Trigger

IviDmmSoftwareTrigger Capability Group

This VI sends a command to trigger the instrument. You must satisfy certain conditions before you can successfully call this VI, The [IviDmm Trigger Source](#) or the [IviDmm Sample Trigger](#) property is set to IVIDMM_VAL_SOFTWARE_TRIG.

If this condition is not true, this VI does not send the software trigger and returns the error IVIDMM_ERROR_TRIGGER_NOT_SOFTWARE.

This VI does not check the instrument status. Typically, the end-user calls this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. The end-user uses the low-level VIs to optimize one or more aspects of interaction with the instrument. To check the instrument status, call the [IviDmm Error-Query](#) VI at the conclusion of the sequence.



 **instrument handle** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviDmm Initialize](#) or [IviDmm Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviDmm Properties

Expand this book to view an alphabetized list of IviDmm properties.

IviDmm AC Maximum Frequency [AC]

IviDmm AC Measurement Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Up	IviDmm Configure AC Bandwidth

Description

Specifies the maximum frequency component of the input signal for AC measurements. The value of this property affects instrument behavior only when you set the [IviDmm Function](#) property to an AC measurement.

Property Node Path

IviDmm»AC Measurements»AC Maximum [AC]

IviDmm AC Minimum Frequency [AC]

IviDmm AC Measurement Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Down	IviDmm Configure AC Bandwidth

Description

Specifies the minimum frequency component of the input signal for AC measurements. The value of this property affects instrument behavior only when you set the [IviDmm Function](#) property to an AC measurement.

Property Node Path

IviDmm»AC Measurements»AC Minimum Frequency [AC]

IviDmm Aperture Time [DI]

IviDmmDeviceInfo Property [DI]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	IviDmm Get Aperture Time Info

Description

Returns the measurement aperture time for the current configuration. The units for this property are either seconds or powerline cycles (PLCs) and are determined by the value of the [IviDmm Aperture Time Units \[DI\]](#) property.

Property Node Path

IviDmm»Configuration Information»Aperture Time [DI]

IviDmm Aperture Time Units [DI]

IviDmmDeviceInfo Property [DI]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	IviDmm Get Aperture Time Info

Description

Returns the units for the property [IviDmm Aperture Time \[DI\]](#).

Property Node Path

IviDmm»Configuration Information»Aperture Time Units [DI]

Defined Values

[IVIDMM_VAL_SECONDS](#)

[IVIDMM_VAL_POWER_LINE_CYCLES](#)

IviDmm Auto Range Value

AutoRangeValue Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	IviDmm Get Auto Range Value

Description

Always returns the actual range the DMM is currently using, even when the DMM is auto-ranging.

Because the value of this property can change as the DMM auto-ranges, you should never cache this property.

Property Node Path

IviDmm»Configuration Information»Auto Range Value

IviDmm Auto Zero [AZ]

AutoZero Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Auto Zero Mode

Description

Specifies the auto-zero mode. In general, the auto-zero capability of a DMM normalizes all measurements based on a Zero Reading. When auto zeroing is enabled, the DMM internally disconnects the input signal and takes a Zero Reading. The DMM then subtracts the Zero Reading from the measurement, which prevents offset voltages present in the DMM's input circuitry from affecting measurement accuracy.

Property Node Path

IviDmm»Measurement Operation Options»Auto Zero [AZ]

Defined Values

[IVIDMM_VAL_AUTO_ZERO_OFF](#)

[IVIDMM_VAL_AUTO_ZERO_ON](#)

[IVIDMM_VAL_AUTO_ZERO_ONCE](#)

IviDmm Fixed Reference Junction [TC]

Thermocouple Property [TC]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Fixed Ref Junction

Description

Specifies the external reference junction temperature when a fixed reference junction type thermocouple is used to take the temperature measurement. The temperature is specified in degrees Celsius. The value of this property affects instrument behavior only when the [IviDmm Reference Junction Type \[TC\]](#) property is set to IVIDMM_VAL_TEMP_REF_JUNC_FIXED.

Property Node Path

IviDmm»Temperature Measurements»Thermocouple»Fixed Reference Junction [TC]

IviDmm Frequency Voltage Range [FRQ]

FrequencyMeasurement Property [FRQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Up	IviDmm Configure Frequency Voltage Range

Description

Specifies the expected maximum value of the input signal for frequency and period measurements. The value of this property affects instrument behavior only when the [IviDmm Function](#) property is set to IVIDMM_VAL_FREQ or IVIDMM_VAL_PERIOD.

Property Node Path

IviDmm»Frequency Measurements»Frequency Voltage Range [FRQ]

IviDmm Function

IviDmm Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Measurement

Description

Specifies the measurement function.

Property Node Path

IviDmm»Basic Operation»Function

Defined Values

[IVIDMM_VAL_DC_VOLTS](#)

[IVIDMM_VAL_AC_VOLTS](#)

[IVIDMM_VAL_DC_CURRENT](#)

[IVIDMM_VAL_AC_CURRENT](#)

[IVIDMM_VAL_2_WIRE_RES](#)

[IVIDMM_VAL_FREQ](#)

[IVIDMM_VAL_PERIOD](#)

[IVIDMM_VAL_4_WIRE_RES](#)

[IVIDMM_VAL_AC_PLUS_DC_VOLTS](#)

[IVIDMM_VAL_AC_PLUS_DC_CURRENT](#)

[IVIDMM_VAL_TEMPERATURE](#)

IviDmm Meas Complete Destination [MP]

Multipoint Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Measurement Complete Destination

Description

After each measurement, the DMM generates a measurement-complete signal. This property specifies the destination of the measurement-complete signal. This signal is commonly referred to as Voltmeter Complete.

Property Node Path

IviDmm»Multipoint Acquisition»Meas Complete Destination [MP]

Defined Values

[IVIDMM_VAL_NONE](#)

[IVIDMM_VAL_EXTERNAL](#)

[IVIDMM_VAL_TTL0](#)

[IVIDMM_VAL_TTL1](#)

[IVIDMM_VAL_TTL2](#)

[IVIDMM_VAL_TTL3](#)

[IVIDMM_VAL_TTL4](#)

[IVIDMM_VAL_TTL5](#)

[IVIDMM_VAL_TTL6](#)

[IVIDMM_VAL_TTL7](#)

[IVIDMM_VAL_ECL0](#)

[IVIDMM_VAL_ECL1](#)

[IVIDMM_VAL_PXI_STAR](#)

[IVIDMM_VAL_RTSI_0](#)

[IVIDMM_VAL_RTSI_1](#)

[IVIDMM_VAL_RTSI_3](#)

IVIDMM_VAL_RTISI_4

IVIDMM_VAL_RTISI_5

IVIDMM_VAL_RTISI_6

IviDmm Powerline Frequency [PLF]

PowerLineFrequency Property [PLF]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Power Line Frequency

Description

Specifies the powerline frequency in hertz.

Property Node Path

IviDmm»Measurement Operation Options»Powerline Frequency [PLF]

IviDmm Range

IviDmm Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A		IviDmm Configure Measurement

Description

Specifies the measurement range. Use positive values to represent the absolute value of the maximum expected measurement. The value is in units appropriate for the current value of the [IviDmm Function](#) property. For example, when you set the IviDmm Function property to `IVIDMM_VAL_DC_VOLTS`, the units are volts. Setting this property to 10.0 configures the DMM to measure DC voltages from -10.0 to $+10.0$ volts.



Note Negative values are reserved for controlling the DMM's auto-ranging capability.

After you set this property to `IVIDMM_VAL_AUTO_RANGE_OFF` or `IVIDMM_VAL_AUTO_RANGE_ONCE`, further queries of this property return the actual range.

When you set this property to `IVIDMM_VAL_AUTO_RANGE_ON`, you can obtain the actual range the DMM is currently using by getting the value of the [IviDmm Auto Range Value \[ARV\]](#) property.

Property Node Path

IviDmm»Basic Operation»Range

Defined Values

[IVIDMM_VAL_AUTO_RANGE_ON](#)

[IVIDMM_VAL_AUTO_RANGE_OFF](#)

[IVIDMM_VAL_AUTO_RANGE_ONCE](#)

IviDmm Reference Junction Type [TC]

Thermocouple Property [TC]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Thermocouple

Description

Specifies the type of reference junction to be used in the reference junction compensation of a thermocouple measurement. The value of this property affects instrument behavior only when the IVIDMM_VAL_TEMP_TRANSUCER_TYPE property is set to IVIDMM_VAL_THERMOCOUPLE.

Property Node Path

IviDmm»Temperature Measurements»Thermocouple»Reference Junction Type [TC]

Defined Values

[IVIDMM_VAL_TEMP_REF_JUNC_INTERNAL](#)

[IVIDMM_VAL_TEMP_REF_JUNC_FIXED](#)

IviDmm Resolution

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Down	IviDmm Configure Measurement

Description

Specifies the measurement resolution of the DMM in absolute units.

Setting this property to lower values increases the measurement accuracy. Setting this property to higher values increases the measurement speed.

Property Node Path

IviDmm»Basic Operation»Resolution

IviDmm RTD Alpha [RTD]

ResistanceTemperatureDevice Property [RTD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure RTD

Description

Specifies the **alpha** parameter for a resistance temperature device. The value of this property affects instrument behavior only when the IVIDMM_VAL_TEMP_TRANSUCER_TYPE property is set to IVIDMM_VAL_2_WIRE_RTD or IVIDMM_VAL_4_WIRE_RTD.

Property Node Path

IviDmm»Temperature Measurements»Resistance Temperature Device»RTD Alpha [RTD]

IviDmm RTD Resistance [RTD]

ResistanceTemperatureDevice Property [RTD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure RTD

Description

Specifies the **R0** parameter (resistance) for a resistance temperature device (RTD). The RTD resistance is also known as the RTD reference value. The value of this property affects instrument behavior only when the `IVIDMM_VAL_TEMP_TRANSDUCER_TYPE` property is set to `IVIDMM_VAL_2_WIRE_RTD` or `IVIDMM_VAL_4_WIRE_RTD`.

Property Node Path

IviDmm»Temperature Measurements»Resistance Temperature Device»RTD Resistance [RTD]

IviDmm Sample Count [MP]

Multipoint Property [MP]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Multipoint

Description

Specifies the number of measurements the DMM takes each time it receives a trigger.

Property Node Path

IviDmm»Multipoint Acquisition»Sample Count [MP]

IviDmm Sample Interval [MP]

Multipoint Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Up	IviDmm Configure Multipoint

Description

Specifies the interval between samples in seconds. This property affects instrument operation only when the [IviDmm Sample Count \[MP\]](#) property is greater than 1 and the [IviDmm Sample Trigger \[MP\]](#) property is set to IVIDMM_VAL_INTERVAL. In this case, the DMM waits between measurements for the length the time you specify with this property.

Property Node Path

IviDmm»Multipoint Acquisition»Sample Interval [MP]

IviDmm Sample Trigger [MP]

Multipoint Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Multipoint

Description

Specifies the sample trigger source. This property affects instrument operation only when the [IviDmm Sample Count \[MP\]](#) property is greater than 1.

When the DMM takes a measurement and the [IviDmm Sample Count \[MP\]](#) property is greater than 1, the DMM does not take the next measurement until the event you specify with this property occurs.

When you set this property to `IVIDMM_VAL_IMMEDIATE`, the DMM does not wait for a trigger of any kind between measurements. When you set this property to `IVIDMM_VAL_EXTERNAL`, the DMM waits for a trigger on the external trigger input before it takes the next measurement. When you set this property to `IVIDMM_VAL_SOFTWARE_TRIG`, the DMM waits until you call the [IviDmm Send Software Trigger \[SWT\]](#) VI before it takes the next measurement. When you set this property to `IVIDMM_VAL_INTERVAL`, the DMM waits the length of time you specify with the [IviDmm Sample Interval \[MP\]](#) property before it takes the next measurement.

Property Node Path

IviDmm»Multipoint Acquisition»Sample Trigger [MP]

Defined Values

[IVIDMM_VAL_IMMEDIATE](#)

[IVIDMM_VAL_EXTERNAL](#)

[IVIDMM_VAL_SOFTWARE_TRIG](#)

[IVIDMM_VAL_INTERVAL](#)

[IVIFGEN_VAL_TTL0](#)

[IVIFGEN_VAL_TTL1](#)

[IVIFGEN_VAL_TTL2](#)

[IVIFGEN_VAL_TTL3](#)

IVIFGEN_VAL_TTL4
IVIFGEN_VAL_TTL5
IVIFGEN_VAL_TTL6
IVIFGEN_VAL_TTL7
IVIFGEN_VAL_ECL0
IVIFGEN_VAL_ECL1
IVIFGEN_VAL_PXI_STAR
IVIFGEN_VAL_RTSI_0
IVIFGEN_VAL_RTSI_1
IVIFGEN_VAL_RTSI_2
IVIFGEN_VAL_RTSI_3
IVIFGEN_VAL_RTSI_4
IVIFGEN_VAL_RTSI_5
IVIFGEN_VAL_RTSI_6

IviDmm Thermistor Resistance [THM]

Thermistor Property [THM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Thermistor

Description

Specifies the resistance of the thermistor in Ohms. The value of this property affects instrument behavior only when the [IviDmm Transducer Type \[TMP\]](#) property is set to IVIDMM_VAL_THERMISTOR.

Property Node Path

IviDmm»Temperature Measurements»Thermistor»Thermistor Resistance [THM]

IviDmm Thermocouple Type [TC]

Thermocouple Property [TC]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Thermocouple

Description

Specifies the type of thermocouple used to measure the temperature. The value of this property affects instrument behavior only when the IVIDMM_VAL_TEMP_TRANSUCER_TYPE is set to IVIDMM_VAL_THERMOCOUPLE.

Property Node Path

IviDmm»Temperature Measurements»Thermocouple»Thermocouple Type [TC]

Defined Values

[IVIDMM_VAL_TEMP_TC_B](#)

[IVIDMM_VAL_TEMP_TC_C](#)

[IVIDMM_VAL_TEMP_TC_D](#)

[IVIDMM_VAL_TEMP_TC_E](#)

[IVIDMM_VAL_TEMP_TC_G](#)

[IVIDMM_VAL_TEMP_TC_J](#)

[IVIDMM_VAL_TEMP_TC_K](#)

[IVIDMM_VAL_TEMP_TC_N](#)

[IVIDMM_VAL_TEMP_TC_R](#)

[IVIDMM_VAL_TEMP_TC_S](#)

[IVIDMM_VAL_TEMP_TC_T](#)

[IVIDMM_VAL_TEMP_TC_U](#)

[IVIDMM_VAL_TEMP_TC_V](#)

IviDmm Transducer Type [TMP]

TemperatureMeasurement Property [TMP]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Transducer Type

Description

Specifies the device used to measure the temperature. The value of this property affects instrument behavior only when the [IviDmm Function](#) property is set to IVIDMM_VAL_TEMPERATURE.

Property Node Path

IviDmm»Temperature Measurements»Transducer Type [TMP]

Defined Values

[IVIDMM_VAL_THERMOCOUPLE](#)

[IVIDMM_VAL_THERMISTOR](#)

[IVIDMM_VAL_2_WIRE_RTD](#)

[IVIDMM_VAL_4_WIRE_RTD](#)

IviDmm Trigger Count [MP]

Multipoint Property [MP]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Multipoint

Description

Specifies the number of triggers the DMM accepts before it returns to the Idle state.

Property Node Path

IviDmm»Multipoint Acquisition»Trigger Count [MP]

IviDmm Trigger Delay

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	See note below.	IviDmm Configure Trigger

Description

Specifies the length of time the DMM waits after it receives the trigger and before it takes a measurement. Use positive values to set the trigger delay in seconds.



Note Negative values are reserved for configuring the DMM to calculate the trigger delay automatically.

Setting this property to `IVIDMM_VAL_AUTO_DELAY_ON` configures the DMM to calculate the trigger delay before each measurement. Setting this property to `IVIDMM_VAL_AUTO_DELAY_OFF` stops the DMM from calculating the trigger delay and sets the trigger delay to the last automatically calculated value.

Property Node Path

IviDmm»Basic Operation»Trigger Delay

Defined Values

[IVIDMM_VAL_AUTO_DELAY_ON](#)

[IVIDMM_VAL_AUTO_DELAY_OFF](#)

IviDmm Trigger Slope [TS]

TriggerSlope Property [TS]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Trigger Slope

Description

Specifies whether the trigger occurs on the rising edge (IVIDMM_VAL_POSITIVE) or on the falling edge (IVIDMM_VAL_NEGATIVE) of the external trigger source.

Property Node Path

IviDmm»Trigger»Trigger Slope [TS]

Defined Values

[IVIDMM_VAL_POSITIVE](#)

[IVIDMM_VAL_NEGATIVE](#)

IviDmm Trigger Source

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviDmm Configure Trigger

Description

Specifies the trigger source. After you call either the Read or Initiate VI, the DMM waits for the trigger you specify with this property. After it receives the trigger, the DMM waits the length of time you specify with the [IviDmm Trigger Delay](#) property. The DMM then takes a measurement.

Property Node Path

IviDmm»Trigger»Trigger Source

Defined Values

[IVIDMM_VAL_IMMEDIATE](#)

[IVIDMM_VAL_EXTERNAL](#)

[IVIDMM_VAL_SOFTWARE_TRIG](#)

[IVIDMM_VAL_TTL0](#)

[IVIDMM_VAL_TTL1](#)

[IVIDMM_VAL_TTL2](#)

[IVIDMM_VAL_TTL3](#)

[IVIDMM_VAL_TTL4](#)

[IVIDMM_VAL_TTL5](#)

[IVIDMM_VAL_TTL6](#)

[IVIDMM_VAL_TTL7](#)

[IVIDMM_VAL_ECL0](#)

[IVIDMM_VAL_ECL1](#)

[IVIDMM_VAL_PXI_STAR](#)

[IVIDMM_VAL_RTSI_0](#)

[IVIDMM_VAL_RTSI_1](#)

[IVIDMM_VAL_RTSI_3](#)

IVIDMM_VAL_RTISI_4

IVIDMM_VAL_RTISI_5

IVIDMM_VAL_RTISI_6

IviDmm Defined Values

Expand this book to view an alphabetized list of IviDmm defined values.

IVIDMM_VAL_2_WIRE_RES

Sets the DMM to measure 2-wire resistance.

Defined value: 5

IVIDMM_VAL_2_WIRE_RTD

Sets the DMM to measure temperature using a 2-wire temperature resistance device. Use the IviDmmResistanceTemperatureDevice extension group to configure additional settings for this transducer type.

Defined Value: 3

IVIDMM_VAL_4_WIRE_RES

Sets the DMM to measure 4-wire resistance.

Defined Value: 101

IVIDMM_VAL_4_WIRE_RTD

Sets the DMM to measure temperature using a 4-wire temperature resistance device. Use the IviDmmResistanceTemperatureDevice extension group to configure additional settings for this transducer type.

Defined Value: 4

IVIDMM_VAL_AC_CURRENT

Sets the DMM to measure AC current.

Defined value: 4

IVIDMM_VAL_AC_PLUS_DC_CURRENT

Sets the DMM to measure AC plus DC current.

Defined Value: 107

IVIDMM_VAL_AC_PLUS_DC_VOLTS

Sets the DMM to measure AC plus DC voltage.

Defined Value: 106

IVIDMM_VAL_AC_VOLTS

Sets the DMM to measure AC voltage.

Defined Value: 2

IVIDMM_VAL_AUTO_DELAY_OFF

Stops the DMM from calculating the trigger delay. Sets the trigger delay to the last trigger delay the DMM calculated.

Defined Value: -2.0

IVIDMM_VAL_AUTO_DELAY_ON

Configures the DMM to calculate the trigger delay before each measurement.

Defined Value: -1.0

IVIDMM_VAL_AUTO_RANGE_OFF

Disables auto-ranging. The DMM sets the range to the value it most recently calculated.

Defined value: -2.0

IVIDMM_VAL_AUTO_RANGE_ON

Configures the DMM to calculate the range before each measurement automatically.

Defined value: -1.0

IVIDMM_VAL_AUTO_RANGE_ONCE

Configures the DMM to calculate the range before the next measurement. The DMM uses this range value for all subsequent measurements.

Defined value: -3.0

IVIDMM_VAL_AUTO_ZERO_OFF

Disables the auto-zero feature.

Defined Value: 0

IVIDMM_VAL_AUTO_ZERO_ON

Configures the DMM to take a Zero Reading for each measurement. The DMM subtracts the Zero Reading from the value it measures.

Defined Value: 1

IVIDMM_VAL_AUTO_ZERO_ONCE

Configures the DMM to take a Zero Reading immediately. The DMM then subtracts this Zero Reading from all subsequent values it measures.

Defined Value: 2

IVIDMM_VAL_DC_CURRENT

Sets the DMM to measure DC current.

Defined value: 3

IVIDMM_VAL_DC_VOLTS

Sets the DMM to measure DC voltage.

Defined value: 1

IVIDMM_VAL_ECL0

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on ECL0.

Defined Value: 119

IVIDMM_VAL_ECL1

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on ECL1.

Defined Value: 120

IVIDMM_VAL_EXTERNAL

Routes the measurement complete signal to the external connector.

Defined Value: 2

IVIDMM_VAL_EXTERNAL (for MEAS_COMPLETE_DEST)

Routes the measurement-complete signal to the external connector.

Defined Value: 2

IVIDMM_VAL_EXTERNAL (for SAMPLE_TRIGGER)

The DMM exits the Wait-For-Sample-Trigger state when a trigger occurs on the external trigger input.

Defined Value: 2

IVIDMM_VAL_EXTERNAL (for TRIGGER_SOURCE)

The DMM exits the Wait-For-Trigger state when a trigger occurs on the external trigger input.

Defined Value: 2

IVIDMM_VAL_EXTERNAL for IviDmm Meas Complete Destination

Routes the measurement-complete signal to the external connector.

Defined Value: 2

IVIDMM_VAL_EXTERNAL for IviDmm Trigger Source

The DMM exits the Wait-For-Trigger state when a trigger occurs on the external trigger input.

Defined Value: 2

IVIDMM_VAL_FREQ

Sets the DMM to measure frequency.

Defined Value: 104

IVIDMM_VAL_IMMEDIATE

The DMM exits the Wait-For-Trigger state or the Wait-For-Sample-Trigger state immediately after entering. It does not wait for a trigger of any kind.

Defined Value: 1

IVIDMM_VAL_INTERVAL

The DMM exits the Wait-For-Sample-Trigger state when the length of time specified by the IVIDMM_ATTR_SAMPLE_INTERVAL property elapses.

Defined Value: 10

IVIDMM_VAL_NEGATIVE

Sets the trigger event to occur on the falling edge of the trigger pulse.

Defined Value: 1

IVIDMM_VAL_NONE

The measurement complete signal is not routed.

Defined Value: -1

IVIDMM_VAL_PERIOD

Sets the DMM to measure period.

Defined Value: 105

IVIDMM_VAL_POSITIVE

Sets the trigger event to occur on the rising edge of the trigger pulse.

Defined Value: 0

IVIDMM_VAL_POWER_LINE_CYCLES

Reports that the units for the value returned by [IVIDMM_ATTR_APERTURE_TIME](#) are Power Line Cycles.

Defined Value: 1

IVIDMM_VAL_PXI_STAR

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on the PXI Star trigger bus.

Defined Value: 131

IVIDMM_VAL_RTSI_0

Routes the measurement complete signal to RTSI0.

Defined Value: 140

IVIDMM_VAL_RTSI_1

Routes the measurement complete signal to RTSI1.

Defined Value: 141

IVIDMM_VAL_RTISI_2

Routes the measurement complete signal to RTISI2.

Defined Value: 142

IVIDMM_VAL_RTISI_3

Routes the measurement complete signal to RTSI3.

Defined Value: 143

IVIDMM_VAL_RTISI_4

Routes the measurement complete signal to RTSI4.

Defined Value: 144

IVIDMM_VAL_RT5I_5

Routes the measurement complete signal to RT5I5.

Defined Value: 145

IVIDMM_VAL_RTSI_6

Routes the measurement complete signal to RTSI6.

Defined Value: 146

IVIDMM_VAL_SECONDS

Reports that the units for the value returned by IVIDMM_ATTR_APERTURE_TIME are seconds.

Defined Value: 0

IVIDMM_VAL_SOFTWARE_TRIG

The DMM exits the Wait-For-Trigger state or the Wait-For-Sample-Trigger state when the [IviDmm_SendSoftwareTrigger](#) function executes.

If you set the IVIDMM_ATTR_TRIGGER_SOURCE property to this value, the instrument can only be triggered by calling the IviDmm_SendSoftwareTrigger function.

If you set the IVIDMM_ATTR_SAMPLE_TRIGGER property to this value, the instrument can only be triggered on each sample by calling the IviDmm_SendSoftwareTrigger function.

Defined Value: 3

IVIDMM_VAL_TEMPERATURE

Sets the DMM to measure temperature in Celsius.

Defined Value: 108

IVIDMM_VAL_TEMP_F

Sets the DMM to measure temperature in Fahrenheit.

Defined Value: 109

IVIDMM_VAL_TEMP_REF_JUNC_FIXED

Sets the DMM to use a fixed value for the thermocouple junction compensation. Use the [IVIDMM_ATTR_TEMP_TC_FIXED_REF_JUNC](#) property to set the fixed reference junction value.

Defined Value: 2

IVIDMM_VAL_TEMP_REF_JUNC_INTERNAL

Sets the DMM to use an internal sensor at the thermocouple junction for the junction compensation.

Defined Value: 1

IVIDMM_VAL_TEMP_TC_B

Sets the DMM to measure temperature from a B-type thermocouple.

Defined Value: 1

IVIDMM_VAL_TEMP_TC_C

Sets the DM7M to measure temperature from a C-type thermocouple.

Defined Value: 2

IVIDMM_VAL_TEMP_TC_D

Sets the DMM to measure temperature from a D-type thermocouple.

Defined Value: 3

IVIDMM_VAL_TEMP_TC_E

Sets the DMM to measure temperature from an E-type thermocouple.

Defined Value: 4

IVIDMM_VAL_TEMP_TC_G

Sets the DMM to measure temperature from a G-type thermocouple.

Defined Value: 5

IVIDMM_VAL_TEMP_TC_J

Sets the DMM to measure temperature from a J-type thermocouple.

Defined Value: 6

IVIDMM_VAL_TEMP_TC_K

Sets the DMM to measure temperature from a K-type thermocouple.

Defined Value: 7

IVIDMM_VAL_TEMP_TC_N

Sets the DMM to measure temperature from an N-type thermocouple.

Defined Value: 8

IVIDMM_VAL_TEMP_TC_R

Sets the DMM to measure temperature from an R-type thermocouple.

Defined Value: 9

IVIDMM_VAL_TEMP_TC_S

Sets the DMM to measure temperature from an S-type thermocouple.

Defined Value: 10

IVIDMM_VAL_TEMP_TC_T

Sets the DMM to measure temperature from a T-type thermocouple.

Defined Value: 11

IVIDMM_VAL_TEMP_TC_U

Sets the DMM to measure temperature from a U-type thermocouple.

Defined Value: 12

IVIDMM_VAL_TEMP_TC_V

Sets the DMM to measure temperature from a V-type thermocouple.

Defined Value: 13

IVIDMM_VAL_THERMISTOR

Sets the DMM to measure temperature using a thermistor. Use the [IviDmmThermistor extension group](#) to configure additional settings for this transducer type.

Defined Value: 2

IVIDMM_VAL_THERMOCOUPLE

Sets the DMM to measure temperature using a thermocouple. Use the [IviDmmThermocouple extension group](#) to configure additional settings for this transducer type.

Defined Value: 1

IVIDMM_VAL_TTL0

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL0.

Defined Value: 112

IVIDMM_VAL_TTL1

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL1.

Defined Value: 111

IVIDMM_VAL_TTL2

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL2.

Defined Value: 113

IVIDMM_VAL_TTL3

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL3.

Defined Value: 114

IVIDMM_VAL_TTL4

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL4.

Defined Value: 115

IVIDMM_VAL_TTL5

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL5.

Defined Value: 116

IVIDMM_VAL_TTL6

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL6.

Defined Value: 117

IVIDMM_VAL_TTL7

The DMM exits the Wait-For-Trigger or the Wait-For-Sample-Trigger state when it receives a trigger on TTL7.

Defined Value: 118

Properties for Controlling IviDmm Simulation



Note You can set Simulation Attributes in MAX on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** or in the simulation interactive panels. Refer to **National Instruments IVI Driver Help»Configuring Your System** for more information.

Two sets of properties exist for use with the IviDmm Simulation driver. The following table describes properties that control behavior of the driver. Another table lists [properties that simulate the status](#) of specific driver VIs.

An [IviDmm Simulator Setup Dialog Box](#) exists to help you configure the measurement simulation for the IviDmm Simulator driver.

Properties for Controlling IviDmm Simulation

Name	Data Type	Access	Applies to	Description
INTERACTIVE_SIMULATION	Boolean	W	No	Specifies whether interactive mode simulation drive panels for getting Non-interactive not want the interactive your test program
SELF_TEST_CODE	Integer	W	No	Specifies the self IviDmm Self-Test result parameter
SELF_TEST_MSG	String	W	No	Specifies the self IviDmm Self-Test message parameter
ERROR_QUERY_CODE	Integer	W	No	Specifies the error IviDmm Error-Code code parameter
ERROR_QUERY_MSG	String	W	No	Specifies the error the IviDmm Error error message string.
DRIVER_REV_QUERY	String	W	No	Specifies the instrument message. When VI is called, the parameter returns
INSTR_REV_QUERY	String	W	No	Specifies the firm When the IviDmm called, the firm returns this string
READING_BASE	Double	W	No	Specifies the base generated measurement measurements value plus the value

				<p>NISDMM_ATTEN and this value multiplied by NISDMM_ATTEN.</p> <p> Note This NISDMM_ATTEN property is</p>
READING_NOISE	Double	W	No	<p>Specifies the relative error generated measurement value of the NISDMM_ATTEN plus this value multiplied by NISDMM_ATTEN minus the value of NISDMM_ATTEN.</p> <p> Note This NISDMM_ATTEN property is</p>
FORCE_OVERRANGE	Boolean	W	No	<p>Specifies whether to force a value for all current measurements. If set to true, measurement values are forced to over-load condition (IVI_VAL_NAN) and an overrange warning is generated.</p>
SIMULATE_STATUS_CODE	Boolean	W	No	<p>Specifies whether to simulate from the IviDmm IviDmm Property Table.</p>

Related Topics

[Setting Up Simulated Errors](#)

[Setting Up Simulation for VXIplug&play VIs](#)

IviDmm Simulator Setup Dialog Box

Two sets of properties exist for use with the IviDmm Simulation Driver. You use the dialog box shown below to configure the measurement simulation for the IviDmm Simulator driver. Scroll down to see a description of each control.



Always prompt for output data simulation

Leaving this control selected causes the panel to appear at every measurement acquisition event in your program. All simulated acquisitions in the program use the same measurement configuration information. Unselecting this control causes the measurement simulation panel to never appear again during the course of your program.

Force Over-Range

Select this control if you want to force an overrange condition to occur during measurement events. When enabled, all measurements simulate a measurement over-load condition by returning Not a Number (Nan) value as the measurement and an overrange warning as the return status.

Measurement Base

Enter the measurement base to use for all computer generated measurements. All computer generated measurement values fall within the range of this value plus or minus the value of the Measurement Noise control. Valid Range: any value

Measurement Noise

Enter the reading noise to use for all computer generated measurements. All computer generated measurement values fall within the range of the value of the Measurement Base control plus or minus this value. Valid Range: any positive value

Properties for Status Simulation in IviDmm

When an IviDmm driver VI is called, the driver queries whether [NISDMM_ATTR_SIMULATE_STATUS_CODE](#) is enabled. If enabled, the driver gets the appropriate VI's simulation status code. If the value of the simulation status code represents a warning (has a positive value), then it is returned as the VI's return status only if no other error or warning occurred before the VI ends. If the value of the simulation status code represents an error (has a negative value), then it is returned as the VI's return status only if no other error occurred before the VI ends.

The following IviDmm VIs do not support status code simulation:

- IviDmm Get Error
- IviDmm Clear Error
- IviDmm Get Next Interchange Warning
- IviDmm Reset Next Interchange Warning
- IviDmm Clear Next Interchange Warning
- IviDmm Get Next Coercion Record

Each VI supported by the IviDmm class driver has a corresponding property that determines the status code to return when status code simulation is enabled. The following table lists the status code properties and the VI for which they return a value. These properties are all of type `ViInt32`, non-readable, and non-channel based.



Note In the following table, the literal string `NISDMM_ATTR_` precedes all property names.

IviDmm Properties for Status Simulation

Property	VI
INIT_STATUS	IviDmm Initialize , IviDmm Initialize With Options
CLOSE_STATUS	IviDmm Close
RESET_STATUS	IviDmm Reset
SELF_TEST_STATUS	IviDmm Self-Te
ERROR_QUERY_STATUS	IviDmm Error-

	Query
ERROR_MESSAGE_STATUS	IviDmm Error Message
REVISION_QUERY_STATUS	IviDmm Revision Query
RESET_DEFAULT_STATUS	IviDmm Reset With Defaults
DISABLE_STATUS	IviDmm Disable
INVALIDATE_STATUS	IviDmm Invalidate All Attributes
CONFIGURE_MEASUREMENT_STATUS	IviDmm Configure Measurement
CONFIGURE_TRIGGER_STATUS	IviDmm Configure Trigger
READ_STATUS	IviDmm Read
FETCH_STATUS	IviDmm Fetch
ABORT_STATUS	IviDmm Abort
INITIATE_STATUS	IviDmm Initiate
IS_OVER_RANGE_STATUS	IviDmm Is Over Range
CONFIGURE_AC_BANDWIDTH_STATUS	IviDmm Configure AC Bandwidth
CONFIGURE_FREQUENCY_VOLTAGE_RANGE_STATUS	IviDmm Configure Frequency Voltage Range
CONFIGURE_TRANSDUCER_TYPE_STATUS	IviDmm Configure Transducer Type
CONFIGURE_FIXED_REF_JUNCTION_STATUS	IviDmm

	Configure Fixed Ref Junction
CONFIGURE_THERMOCOUPLE_STATUS	IviDmm Configure Thermocouple
CONFIGURE_RTD_STATUS	IviDmm Configure RTD
CONFIGURE_THERMISTOR_STATUS	IviDmm Configure Thermistor
CONFIGURE_MEAS_COMPLETE_DEST_STATUS	IviDmm Configure Measurement Complete Destination
CONFIGURE_MULTI_POINT_STATUS	IviDmm Configure MultiPoint
READ_MULTI_POINT_STATUS	IviDmm Read MultiPoint
FETCH_MULTI_POINT_STATUS	IviDmm Fetch MultiPoint
CONFIGURE_TRIGGER_SLOPE_STATUS	IviDmm Configure Trigger Slope
SEND_SOFTWARE_TRIGGER_STATUS	IviDmm Send Software Trigger
GET_APERTURE_TIME_INFO_STATUS	IviDmm Get Aperture Time Info
GET_AUTO_RANGE_VALUE_STATUS	IviDmm Get Auto Range Value
CONFIGURE_AUTO_ZERO_MODE_STATUS	IviDmm Configure Auto Zero Mode
CONFIGURE_POWER_LINE_FREQUENCY_STATUS	IviDmm

Configure Power
Line Frequency

□ IviFgen Class Driver Reference for LabVIEW

Concepts

[IviFgen Class Driver Overview](#)

[IviFgen Extension Capability Groups](#)

[IviFgen Behavior Models](#)

IviFgen VI and Property Reference

[IviFgen VI Tree](#)

[IviFgen Properties by Group](#)

[IviFgen Property Value Definitions](#)

[IviFgen Error and Completion Codes](#)

Other Reference

[Interchangeability Checking](#)

[Simulation](#)

IviFgen Class Driver Overview

This instrument driver provides programming support for the IviFgen Class. The IviFgen class driver conceptualizes a function generator as an instrument capable of generating an analog voltage waveform, and can be applied to a wide range of instruments. The output signal is typically functional in nature (for example sinusoidal or square). Some instruments support the generation of arbitrary waveforms, which consist of user-specified data. If the function generator also supports the generation of arbitrary waveform sequences, the output signal can consist of a sequence of repeated arbitrary waveforms. The IviFgen class driver is divided into base capabilities and extensions. The base capabilities configure a function generator for basic signal output. With the extensions, you can configure a function generator to generate a specific type of waveform. An instrument driver must support either the [IviFgenStdFunc](#) or [IviFgenArbWfm](#) extension. The driver contains all the VIs that the IVI Foundation requires for the IviFgen specification. This driver requires NI-VISA and the IVI Compliance Package to be installed.

Use this driver to develop programs that are independent of a particular function generator. You can use this class driver with any function generator that has an IVI instrument specific driver that is compliant with the IviFgen class. The IviFgen class driver accesses the specific driver for your instrument using the configuration information you supply with MAX.

The IviFgen class driver divides instrument capabilities into a [IviFgenBase capability group](#) and multiple extension capability groups. The IviFgenBase group supports function generators capable of configuring output impedances and the reference clock source, and enabling and disabling the function generator's outputs. The base capabilities alone do not support generating an output signal. Instead, a function generator must support at least one of the signal output extensions: IviFgenStdFunc, IviFgenArbWfm, or IviFgenArbSeq. The base group consists of the following VIs and properties. These VIs and properties are implemented by all function generator instrument drivers that are compliant with the IviFgen class.

IviFgen Extension Capability Groups

The IviFgen class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Abbreviation	Description
IviFgenStdFunc	[STD]	For instruments capable of generating standard periodic waveforms.
IviFgenArbWfm	[ARB]	For instruments capable of generating user-defined arbitrary waveforms.
IviFgenArbFrequency	[AF]	For instruments capable of specifying the rate at which the function generator produces an entire cycle of an arbitrary waveform.
IviFgenArbSeq	[SEQ]	For instruments capable of generating a user-defined sequence of arbitrary waveforms.
IviFgenTrigger	[TRG]	For instruments capable of producing a signal upon receipt of a trigger.
IviFgenInternalTrigger	[IT]	For instruments capable of producing a signal upon receipt of a trigger from an internal trigger source.
IviFgenSoftwareTrigger	[SWT]	For instruments capable of producing a signal upon receipt of a software trigger.
IviFgenBurst	[BST]	For instruments capable of generating a fixed number of waveform cycles signal upon

		receipt of a trigger.
IviFgenModulateAM	[AM]	For instruments capable of applying amplitude modulation to an output signal.
IviFgenModulateFM	[FM]	For instruments capable of applying frequency modulation to an output signal.

Related Topics

[IVI Inherent Class Capabilities](#)

[Standard Function Waveform Description Overview](#)

[Arbitrary Waveform Overview](#)

[Arbitrary Sequence Overview](#)

[Amplitude Modulation Overview](#)

[Frequency Modulation Overview](#)

[Output Stage Overview](#)

Standard Function Waveform Description

Overview

Instrument vendors typically have different definitions for waveform properties. In order to achieve a consistent waveform description between different instrument vendors, this class driver provides waveform property definitions that must be followed when developing instrument drivers. The definitions for these waveform properties are as follows:

Waveform Type—The overall "shape" of one period of the standard waveform. This specification defines six waveform types: Sine, Square, Triangle, Ramp Up, Ramp Down, and DC.

Amplitude—The difference between the maximum and minimum waveform values, or the peak-to-peak voltage value.

DC Offset—The difference between the average of the maximum and minimum waveform values and the x-axis (0 volts). A positive DC offset places the middle of the waveform above the x-axis, while a negative DC offset places the middle of the waveform below the x-axis.

Frequency—The number of waveform cycles generated in one second.

Start Phase—Specifies the waveform's horizontal offset. The units are degrees of one waveform cycle. For example, a 180 degree phase offset means output generation begins half way through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle. It is therefore identical to a start phase of 0 degrees.

The following illustration diagrams these properties.



Illustration of Basic Waveform Properties

Duty Cycle—A square waveform requires an additional parameter to configure the duty cycle of the waveform. Duty cycle is defined as the percentage of time during one cycle for which the square wave is at its high value.



Square Waveform with 75 percent Duty Cycle

Arbitrary Waveform Overview

The IviFgenArbWfm Extension Group supports function generators capable of producing user-defined arbitrary waveforms. You can modify parameters of the arbitrary waveform such as sample rate, waveform gain, and waveform offset. The IviFgenArbWfm extension group includes VIs for creating, configuring, and generating arbitrary waveforms, and for returning information about arbitrary waveform creation.

This extension affects instrument behavior when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_ARB or IVIFGEN_VAL_OUTPUT_SEQ .

Before a function generator can produce an arbitrary waveform, you must configure some signal generation properties. This specification provides definitions for arbitrary waveform properties that must be followed when developing instrument drivers. The definition of an arbitrary waveform and its properties are given in the following list:

- **Arbitrary Waveform**—A user-defined series of sequential data points, between -1.0 and 1.0 inclusive, that describe an output waveform.
- **Gain**—The factor by which the function generator scales the arbitrary waveform data. For example, a gain value of 2.0 causes the waveform data to range from -2.0 V to +2.0 V.
- **Offset**—The value the function generator adds to the scaled arbitrary waveform data. For example, scaled arbitrary waveform data that ranges from -1.0 V to +1.0 V is generated from 0.0 V to 2.0 V when the end user specifies a waveform offset of 1.0 V.



Note The offset is added to any inherent offset in the arbitrary waveform data.

The following figures illustrate the definitions for arbitrary waveform properties.



Examples of Normalized, Scaled, and Offset Data



Size, Sample Rate, and Time Elements of a Waveform

The sample rate is the reciprocal of the amount of time for which points in

the arbitrary waveform is generated. The frequency at which the function generator produces one cycle of an arbitrary waveform can be expressed by the equation:

$$frequency = \frac{numPts}{sampleRate}$$

Arbitrary Sequence Overview

In addition to allowing you to specify an arbitrary waveform, some function generators allow you to specify a sequence of arbitrary waveforms for the function generator to produce. This class driver defines an arbitrary sequence as a list of arbitrary waveforms to produce. Each waveform in the sequence is repeated a discrete number of times before producing the next waveform. When generating an arbitrary sequence, the waveform properties of Gain, Offset, and Sample Rate, defined in the [Arbitrary Waveform Overview](#) apply to all waveforms in the sequence. The following figure illustrates the definition of an arbitrary sequence.



Waveform Sequencing

Amplitude Modulation Overview

Amplitude modulation is accomplished by varying the amplitude of a carrier waveform according to the amplitude of a modulating waveform. The general equation for applying amplitude modulation to a waveform is:

$$AM(t) = [M(t) + 1] \times C(t),$$

where $C(t)$ is the carrier waveform, $M(t)$ is the modulating waveform, and $AM(t)$ is the modulated signal.

This class driver provides modulating waveform property definitions that must be followed when developing specific instrument drivers. The carrier waveform is defined as the waveform that the function generator produces without any modulation. The modulating waveform is defined by the following properties:

Waveform Type—The overall "shape" of one period of the modulating waveform. This class driver defines five modulating waveform types: Sine, Square, Triangle, Ramp Up, and Ramp Down.

Frequency—The number of modulating waveform cycles generated in one second.

Modulation Depth—The extent to which the modulating waveform affects the amplitude of the carrier waveform. This value is expressed as a percentage of total modulation.

At the maximum peak of the modulating waveform, the amplitude of the output signal is equal to $(100.0 + \textit{Modulation Depth})$ percent of the carrier signal amplitude. At the minimum peak of the modulating waveform, the amplitude of the output signal is equal to $(100.0 - \textit{Modulation Depth})$ percent of the carrier signal amplitude. At a modulation depth of 0 percent, the modulating waveform has no effect on the carrier waveform. At a modulation depth of 100 percent, the amplitude of the output signal varies between 0.0 V and twice the amplitude of the carrier signal.

The following diagram illustrates the effect of amplitude modulation on a carrier signal, and the effect on the output signal of varying the modulation depth.



One kHz Carrier Sine Wave and a 100 percent Amplitude Modulated

Wave

Frequency Modulation Overview

Frequency modulation is accomplished by varying the frequency of a carrier waveform according to the amplitude of a modulating waveform. The general equation for a frequency modulated waveform is:

$$FM(t) = C[t + (M(t))],$$

where $C(t)$ is the carrier waveform, $M(t)$ is the modulating waveform, and $FM(t)$ is the frequency modulated signal.

This class driver provides modulating waveform property definitions that must be followed when developing specific instrument drivers. The carrier waveform is defined as the waveform the function generator produces without any modulation. The modulating waveform is defined by the following properties:

- **Waveform Type**

—The overall "shape" of one period of the modulating waveform. This class driver defines five modulation waveform types: Sine, Square, Triangle, Ramp Up, and Ramp Down.

- **Frequency**

—The number of modulating waveform cycles generated in one second.

- **Peak Frequency Deviation**

—The variation of frequency the modulating waveform applies to the carrier waveform. This value is expressed in hertz. At 0 hertz deviation, the modulating waveform has no effect on the carrier waveform. As frequency deviation increases, the frequency variation in the modulated waveform increases.

At the maximum peak of the modulating signal, the frequency of the output signal is equal to the frequency of the carrier signal plus the frequency of the modulating signal. At the minimum peak of the modulating signal, the frequency of the output signal is equal to the frequency of the carrier signal minus the frequency of the modulating signal.

The following diagram illustrates the effect of frequency modulation on a carrier signal, and the effect on the output signal of varying the peak frequency deviation.



One kHz Carrier Sine Wave and a Frequency Modulated Wave at 20 Hz Peak Deviation

Output Stage Overview

This class driver models the output stage of the function generator with the following properties: Output Voltage, Output Impedance, and System Impedance. The definitions of these properties are as follows:

- **Output Voltage**—The voltage of the waveform at the output terminal when connected to a matched termination.
- **Output Impedance**—The impedance of the function generator at the output terminal.
- **System Impedance**—The impedance of the system attached to the output terminal.

These properties are shown in the following illustration.



Illustration of Output Path Properties

The combination of the output impedance of the function generator and the impedance of the system attached to the output terminal of the function generator form a voltage divider network. The actual voltage across the system impedance depends on this voltage divider network. The amplitude of the signal the function generator produces is based on the assumption that the system impedance matches the function generator's output impedance. If the system impedance differs from the output impedance, you must consider this factor to achieve the desired output voltage.

The actual output voltage can be found by using the following equations:

$$\frac{V_o}{R_s} = \frac{V_p - V_o}{R_o}$$

$$V_o = \frac{V_p R_s}{(R_o + R_s)}$$

When the output load matches the system load (in other words, $R_s = R_o$, $V_o = \frac{1}{2}V_p$), the instrument driver calculates the correct value for V_p to result in V_o being set to the requested user value. If you have not matched the system impedance to the output impedance, you must calculate the output voltage on your own. For example, consider the case where R_s is much larger than R_o , as when a very high system impedance is attached

to the output terminal. In this case, $V_o = V_p$. However, because the instrument driver has assumed a matched load (in other words, $V_o = \frac{1}{2}V_p$) the output voltage is twice the requested value. In this case, you should request an output voltage that is half of the desired output value to achieve the intended output.

IviFgen Extension Capability Groups

The IviFgen class driver identifies VIs and properties that belong to an extension group with the following group names. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Abbreviation	Description
IviFgenStdFunc	[STD]	For instruments capable of generating standard periodic waveforms.
IviFgenArbWfm	[ARB]	For instruments capable of generating user-defined arbitrary waveforms.
IviFgenArbFrequency	[AF]	For instruments capable of specifying the rate at which the function generator produces an entire cycle of an arbitrary waveform.
IviFgenArbSeq	[SEQ]	For instruments capable of generating a user-defined sequence of arbitrary waveforms.
IviFgenTrigger	[TRG]	For instruments capable of producing a signal upon receipt of a trigger.
IviFgenInternalTrigger	[IT]	For instruments capable of producing a signal upon receipt of a trigger from an internal trigger source.
IviFgenSoftwareTrigger	[SWT]	For instruments capable of producing a signal upon receipt of a software trigger.
IviFgenBurst	[BST]	For instruments capable of generating a fixed number of waveform cycles signal upon

		receipt of a trigger.
<u>IviFgenModulateAM</u>	[AM]	For instruments capable of applying amplitude modulation to an output signal.
<u>IviFgenModulateFM</u>	[FM]	For instruments capable of applying frequency modulation to an output signal.

IviFgenArbFrequency Extension Group

The IviFgenArbFrequency extension group supports function generators capable of producing arbitrary waveforms that allow you to set the rate at which an entire arbitrary waveform is generated. In order to support this extension, a driver must first support the IviFgenArbWfm extension group. This extension uses the IviFgenArbWfm extension group's properties of waveform handle, gain, and offset to configure an arbitrary waveform.

IviFgenArbFrequency VI

[IviFgen Configure Arb Frequency](#)

IviFgenArbFrequency Property

[IviFgen Arb Frequency \[AF\]](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgenArbSeq Extension Group

The IviFgenArbSeq extension group supports function generators capable of producing sequences of arbitrary waveforms. In order to support this extension, a driver must first support the IviFgenArbWfm extension group. This extension uses the IviFgenArbWfm extension group's properties of sample rate, gain, and offset to configure a sequence. The IviFgenArbSeq extension group includes VIs for creating, configuring, and generating sequences, and for returning information about arbitrary sequence creation.

IviFgenArbSeq VIs

[IviFgen Query Arb Sequence Capabilities](#)

[IviFgen Create Arbitrary Sequence \[SEQ\]](#)

[IviFgen Configure Arbitrary Sequence](#)

[IviFgen Clear Arbitrary Sequence](#)

[IviFgen Clear Arbitrary Memory](#)

IviFgenArbSeq Properties

[IviFgen Arbitrary Sequence Handle](#)

[IviFgen Max Number of Sequences](#)

[IviFgen Max Loop Count](#)

[IviFgen Max Sequence Length](#)

[IviFgen Min Sequence Length](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgenArbWfm Extension Group

The IviFgenArbWfm extension group supports function generators capable of producing user-defined arbitrary waveforms. You can modify parameters of the arbitrary waveform such as sample rate, waveform gain, and waveform offset. The IviFgenArbWfm extension group includes VIs for creating, configuring, and generating arbitrary waveforms, and for returning information about arbitrary waveform creation.

IviFgenArbWfm VIs

[IviFgen Configure Sample Rate](#)

[IviFgen Query Arb Waveform Capabilities](#)

[IviFgen Create Arbitrary Wfm](#)

[IviFgen Configure Arbitrary Wfm](#)

[IviFgen Clear Arbitrary Wfm](#)

IviFgenArbWfm Properties

[IviFgen Arbitrary Waveform Gain](#)

[IviFgen Arbitrary Waveform Offset](#)

[IviFgen Arb Sample Rate](#)

[IviFgen Arbitrary Waveform Handle](#)

[IviFgen Max Number of Waveforms](#)

[IviFgen Max Waveform Size](#)

[IviFgen Min Waveform Size](#)

[IviFgen Waveform Quantum](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgen Base Capability Group

The IviFgen Base capabilities support the most basic function generator capabilities. You can configure the output impedance and reference clock source, and enable or disable the function generator's output connectors. The base capabilities do not support the generation of a specific kind of output signal. Instead, a function generator must support either the [IviFgenStdFunc](#) or [IviFgenArbWfm](#) extensions. This organization is required because many function generators support only one of these extension groups. If a function generator supports more than one of these extensions, the output mode can be configured to specify which extension the function generator uses to produce a signal.

IviFgenBase VIs

[IviFgen Configure Output Mode](#)

[IviFgen Configure Ref Clock Source](#)

[IviFgen Configure Operation Mode](#)

[IviFgen Configure Output Impedance](#)

[IviFgen Configure Output Enabled.](#)

[IviFgen Initiate Generation](#)

[IviFgen Abort Generation](#)

IviFgenBase Properties

[IviFgen Operation Mode](#)

[IviFgen Output Enabled](#)

[IviFgen Output Impedance](#)

[IviFgen Output Mode](#)

[IviFgen Reference Clock Source](#)

For more VI and property information:

[IviFgen VI Tree](#)

[IviFgen Properties](#)

Related Topics

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

[IviFgenBase Behavior Model](#)

IviFgenBurst Extension Group

The IviFgenBurst Extension Group supports function generators capable of producing a discrete number of waveform cycles upon receipt of a trigger. You specify the trigger source with the IviFgenTrigger Extension Group.

For standard and arbitrary waveforms, a cycle is one period of the waveform. For arbitrary sequences, a cycle is one complete progression through the generation of all iterations of all waveforms in the sequence.

IviFgenBurst VI

IviFgen Configure Burst Count

IviFgenBurst Property

[IviFgen Burst Count](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

[IviFgenBurst behavior model](#)

IviFgenInternalTrigger Extension Group

The IviFgenInternalTrigger Extension Group supports function generators capable of generating output based on an internally generated trigger signal. You can configure the rate at which internal triggers are generated.

IviFgenInternalTrigger VI

IviFgen Configure Internal Trigger Rate

IviFgenInternalTrigger Property

[IviFgen Internal Trigger Rate](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

[IviFgenInternalTrigger behavior model](#)

IviFgenModulateAM Extension Group

The IviFgenModulateAM Extension Group supports function generators that can apply amplitude modulation to an output signal. You can enable or disable amplitude modulation, and specify the source of the modulating signal. If the function generator supports an internal modulation source, you can specify the waveform type, frequency, and modulation depth.

IviFgenModulateAM VIs

[IviFgen Configure AM Enabled](#)

[IviFgen Configure AM Source](#)

[IviFgen Configure AM Internal](#)

IviFgenModulateAM Properties

[IviFgen AM Enabled](#)

[IviFgen AM Source](#)

[IviFgen AM Internal Depth](#)

[IviFgen AM Internal Frequency](#)

[IviFgen AM Internal Waveform](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgenModulateFM Extension Group

The IviFgenModulateFM Extension Group supports function generators that can apply frequency modulation to an output signal. You can enable or disable frequency modulation, and specify the source of the modulating signal. If the function generator supports an internal modulation source, you can specify the waveform type, frequency, and peak frequency deviation.

IviFgenModulateFM VIs

[IviFgen Configure FM Enabled](#)

[IviFgen Configure FM Source](#)

[IviFgen Configure FM Internal](#)

IviFgenModulateFM Properties

[IviFgen FM Enabled](#)

[IviFgen FM Source](#)

[IviFgen FM Internal Deviation](#)

[IviFgen FM Internal Frequency](#)

[IviFgen FM Internal Waveform](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgenSoftwareTrigger Extension Group

The IviFgenSoftwareTrigger Extension Group supports function generators that can generate output based on a software trigger signal. You can send a software trigger to cause signal output to occur.

IviFgenSoftwareTrigger VI

[IviFgen Send Software Trigger](#)

Related Topics

[IviFgen VI Tree](#)

IviFgenStdFunc Extension Group

The IviFgenStdFunc extension group supports function generators that can produce manufacturer-supplied periodic waveforms. You can modify parameters of the waveform such as frequency, amplitude, DC offset, and phase offset.

IviFgenStdFunc VI

[IviFgen Configure Standard Waveform](#)

IviFgenStdFunc Properties

[IviFgen Func Amplitude](#)

[IviFgen DC Offset](#)

[IviFgen Duty Cycle High](#)

[IviFgen Frequency](#)

[IviFgen Start Phase](#)

[IviFgen Waveform](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgenTrigger Extension Group

The IviFgenTrigger Extension Group supports function generators capable of responding to a trigger signal. The value of the [IviFgen Operation Mode](#) property determines how the function generator responds to the trigger source.

IviFgenTrigger VI

[IviFgen Configure Trigger Source](#)

IviFgenTrigger Property

[IviFgen Trigger Source \[TRG\]](#)

Related Topics

[IviFgen VI Tree](#)

[IviFgen Properties](#)

IviFgen Behavior Models

[IviFgenArbSeq](#)

[IviFgenArbWfm](#)

[IviFgenBase](#)

[IviFgenBurst](#)

[IviFgenArbFrequency](#)

[IviFgenInternalTrigger](#)

[IviFgenModulateAM](#)

[IviFgenModulateFM](#)

[IviFgenStdFunc](#)

[IviFgenTrigger](#)

[IviFgenSoftwareTrigger](#)

IviFgenArbFrequency Behavior Model

The IviFgenArbFrequency Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenArbFrequency settings.

IviFgenArbSeq Behavior Model

The IviFgenArbSeq Extension Group follows the behavior model of the [IviFgenArbWfm](#) capability group. The only modification to the behavior model from the fundamental capability group is the ability to configure IviFgenArbSeq settings.

IviFgenArbWfm Behavior Model

The IviFgenArbWfm Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenArbWfm settings.

IviFgenBase Behavior Model

The following behavior model shows the relationship between the IviFgen base capability group and function generator behavior.



IviFgen Behavior Model

After you call the [ClassPrefix Initialize](#), [ClassPrefix Initialize With Options](#), or [ClassPrefix Reset](#) VIs, the function generator enters the *output generation* state.

Calling [IviFgen Abort Generation](#) moves the function generator to the *configuration* state. Depending on the specific function generator, this might cause signal generation to cease. Any changes made to the instrument while in the *configuration* state take place no later than when you call [IviFgen Initiate Generation](#), but can occur before then.

Calling [IviFgen Initiate Generation](#) moves the function generator back to the *output generation* state. The output signal then reflects all configuration changes made to the function generator while in the *configuration* state. Any changes made to the instrument while in the *output generation* state take place immediately.

When the function generator is in the *output generation* state, it generates output based on the current value of the [IviFgen Operation Mode](#) property. If the IviFgen Operation Mode property is set to `IVIFGEN_VAL_OPERATE_CONTINUOUS`, the function generator produces the waveform continuously.

If the IviFgen Operation Mode property is set to a value other than `IVIFGEN_VAL_OPERATE_CONTINUOUS`, the properties and VIs of a corresponding extension group control the operation of the function generator. Refer to the behavior model for the corresponding extension group for details regarding instrument behavior.

This specification does not define when the function generator changes to a new operation mode if the value of the IviFgen Operation Mode property changes while the function generator is in the *output generation* state.

IviFgenBurst Behavior Model

The behavior model for the IviFgenBurst extension capability group leverages the behavior model for the IviFgenBase extension group. The only difference is what happens while the function generator is in the *output generation* state. The following diagram shows the *output generation* state for the IviFgenBurst extension group.



IviFgenBurst Behavior Model

When the function generator enters the *output generation* state and the [IviFgen Operation Mode](#) property is set to `IVIFGEN_VAL_OPERATE_BURST`, the function generator enters the *wait-for-trigger* state. The function generator exits the *wait-for-trigger* state when it receives a trigger from the source that the [IviFgen Trigger Source \[TRG\]](#) property specifies.

After the function generator exits the *wait-for-trigger* state it generates one complete cycle of the currently configured waveform. The [IviFgen Output Mode](#) property determines the type of waveform that the function generator produces. The waveform could be a standard VI, an arbitrary waveform, or an arbitrary sequence.

After the function generator produces each waveform cycle, the function generator compares the total number of waveform cycles that have been generated since the function generator exited the *wait-for-trigger* state with the value of the [IviFgen Burst Count](#) property. If the total number of waveforms generated is less than the value of the IviFgen Burst Count property, the function generator produces another waveform cycle. If the total number of waveforms generated is equal to the value of the IviFgen Burst Count property, the function generator returns to the *wait-for-trigger* state.

After the function generator begins to generate waveform cycles, it ignores all trigger signals until the number of waveform cycles specified by the IviFgen Burst Count property have been generated.

IviFgenInternalTrigger Behavior Model

The IviFgenInternalTrigger Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenInternalTrigger settings.

IviFgenModulateAM Behavior Model

The IviFgenModulateAM Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenModulateAM settings.

IviFgenModulateFM Behavior Model

The IviFgenModulateFM Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenModulateFM settings.

IviFgenSoftwareTrigger Behavior Model

The IviFgenSoftwareTrigger Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenSoftwareTrigger settings.

IviFgenStdFunc Behavior Model

The IviFgenStdFunc Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenStdFunc settings.

IviFgenTrigger Behavior Model

The IviFgenTrigger Extension Group follows the behavior model of the [IviFgenBase](#) capability group. The only modification to the behavior model from the IviFgenBase capability group is the ability to configure IviFgenTrigger settings.

IviFgen VI Tree

The VI tree for the IviFgen class driver, including IVI and *VXIplug&play* required VIs, is shown in the following table.



Note You use the LabVIEW Property Node to get and set properties.

IviFgen VI Tree

Name or Class	VI Name	Required By
Initialize	IviFgen Initialize	VPP
Initialize With Options	IviFgen Initialize With Options	IVI
Configuration		
Configure Output Mode	IviFgen Configure Output Mode	IviFgen
Configure Operation Mode	IviFgen Configure Operation Mode	IviFgen
Configure Ref Clock Source	IviFgen Configure Ref Clock Source	IviFgen
Configure Output Impedance	IviFgen Configure Output Impedance	IviFgen
Configure Output Enabled	IviFgen Configure Output Enabled.	IviFgen
Standard Waveform		
Configure Standard Waveform [STD]	IviFgen Configure Standard Waveform	IviFgenStdFunc
Arbitrary Waveform		
Query Arb Waveform Capabilities [ARB]	IviFgen Query Arb Waveform Capabilities	IviFgenArbWfm
Create Arbitrary Waveform [ARB]	IviFgen Create Arbitrary Wfm	IviFgenArbWfm
Configure Sample Rate [ARB]	IviFgen Configure Sample Rate	IviFgenArbWfm
Configure Arbitrary	IviFgen Configure	IviFgenArbWfm

Waveform [ARB]	Arbitrary Wfm	
Clear Arbitrary Waveform [ARB]	IviFgen Clear Arbitrary Wfm	IviFgenArbWfm
Arbitrary Waveform Frequency		
Configure Arb Frequency [ARBF]	IviFgen Configure Arb Frequency	IviFgenArbFrequency
Arbitrary Sequence		
Query Arbitrary Sequence Capabilities [SEQ]	IviFgen Query Arb Sequence Capabilities	IviFgenArbSeq
Create Arbitrary Sequence [SEQ]	IviFgen Create Arbitrary Sequence [SEQ]	IviFgenArbSeq
Configure Arbitrary Sequence [SEQ]	IviFgen Configure Arbitrary Sequence	IviFgenArbSeq
Clear Arbitrary Sequence [SEQ]	IviFgen Clear Arbitrary Sequence	IviFgenArbSeq
Clear Arbitrary Memory [SEQ]	IviFgen Clear Arbitrary Memory	IviFgenArbSeq
Trigger		
Configure Trigger Source [TRG]	IviFgen Configure Trigger Source	IviFgenTrigger
Internal Trigger		
Configure Internal Trigger Rate [ITRG]	IviFgen Configure Internal Trigger Rate	IviFgenInternalTrigger
Burst		
Configure Burst Count [BST]	IviFgen Configure Burst Count	IviFgenBurst

Amplitude Modulation

Configure AM Enabled [AM]	IviFgen Configure AM Enabled	IviFgenModulateAM
Configure AM Source [AM]	IviFgen Configure AM Source	IviFgenModulateAM
Configure AM Internal	IviFgen Configure AM Internal	IviFgenModulateAM

Frequency Modulation

Configure FM Enabled [FM]	IviFgen Configure FM Enabled	IviFgenModulateFM
Configure FM Source [FM]	IviFgen Configure FM Source	IviFgenModulateFM
Configure FM Internal [FM]	IviFgen Configure FM Internal	IviFgenModulateFM

Action VIs

Initiate Generation	IviFgen Initiate Generation	IviFgen
Abort Generation	IviFgen Abort Generation	IviFgen
Send Software Trigger [SWT]	IviFgen Send Software Trigger	IviFgenSoftwareTrigger

Utility VIs

Reset	IviFgen Reset	VPP
Reset With Defaults	Reset With Defaults	VPP
Self-Test	IviFgen Self-Test	VPP
Disable	Disable	VPP
Invalidate All Attributes	Invalidate All Attributes	VPP

Revision Query	IviFgen Revision Query	VPP
Error-Query	IviFgen Error-Query	VPP
Error Message	IviFgen Error Message	VPP
Get Channel Name	IviFgen Get Channel Name	VPP
Interchangeability Info		
Get Next Interchange Warning	IviFgen Get Next Interchange Warning	IVI
Reset Interchange Check	IviFgen Reset Interchange Check	IVI
Clear Interchange Warnings	IviFgenClear Interchange Warnings	IVI
Coercion Info		
Get Next Coercion Record	IviFgen Get Next Coercion Record	IVI
Close	IviFgen Close	VPP

Related Topics

[IviFgen Properties](#)

[IVI Inherent VIs](#)

IviFgen Properties



Note You use the LabVIEW Property Node to get and set properties.

IviFgen Base Properties

[IviFgen Operation Mode](#)

[IviFgen Output Enabled](#)

[IviFgen Output Impedance](#)

[IviFgen Output Mode](#)

[IviFgen Reference Clock Source](#)

IviFgenStdFunc Properties

[IviFgen Func Amplitude](#)

[IviFgen DC Offset](#)

[IviFgen Duty Cycle High](#)

[IviFgen Frequency](#)

[IviFgen Start Phase](#)

[IviFgen Waveform](#)

IviFgenArbWfm Properties

[IviFgen Arbitrary Waveform Gain](#)

[IviFgen Arbitrary Waveform Offset](#)

[IviFgen Arb Sample Rate](#)

[IviFgen Arbitrary Waveform Handle](#)

[IviFgen Max Number of Waveforms](#)

[IviFgen Max Waveform Size](#)

[IviFgen Min Waveform Size](#)

[IviFgen Waveform Quantum](#)

IviFgenArbFrequency Properties

[IviFgen Arb Frequency](#)

IviFgenArbSeq Properties

IviFgen Arbitrary Sequence Handle

IviFgen Max Number of Sequences

IviFgen Max Loop Count

IviFgen Max Sequence Length

IviFgen Min Sequence Length

IviFgenTrigger Properties

[IviFgen Trigger Source](#)

IviFgenInternalTrigger Properties

[IviFgen Internal Trigger Rate](#)

IviFgenBurst Properties

IviFgen Burst Count

IviFgenModulateAM Properties

[IviFgen AM Enabled](#)

[IviFgen AM Source](#)

[IviFgen AM Internal Depth](#)

[IviFgen AM Internal Frequency](#)

IviFgenModulateFM Properties

[IviFgen FM Enabled](#)

[IviFgen FM Source](#)

[IviFgen FM Internal Deviation](#)

[IviFgen FM Internal Frequency](#)

IviFgen Property Value Definitions

The following table defines the ID value for all IviFgen class properties.



Note In the following table the literal string IVIFGEN precedes all property names, and the literal string IVIFGEN_VAL_ precedes all defined value names.

IviFgen Property Defined Values

Property	Value Name	Defined Value	Description
OPERATION MODE	Continuous	0	Configures the function generator generate output continuously.
OPERATE_BURST	Burst	1	Configures the function generator generate a burst of waveform cyc
OUTPUT MODE	Standard Function	0	The driver uses the properties and functions of the IviFgenStdFunc extension group to configure the function generator's output signal
	Arbitrary Waveform	1	The driver uses the properties and functions of the IviFgenArbWfm Extension Group to configure the function generator's output signal
	Arbitrary Sequence	2	The driver uses the properties and functions of the IviFgenArbWfm and IviFgenArbSeq Extension Groups together to configure the function generator's output signal.
REFERENCE CLOCK SOURCE	Internal	0	The function generator produces reference clock signal internally.
	External	1	The function generator receives the reference clock signal from an external source.
	Sine	1	Configures the function generator produce a sinusoid waveform.
	Square	2	Configures the function generator produce a square waveform.
	Triangle	3	Configures the function generator produce a triangular waveform.
	Ramp Up	4	Configures the function generator produce a positive ramp waveform
	Ramp	5	Configures the function generator

	Down		produce a negative ramp waveform
	DC	6	Configures the function generator produce a constant voltage.
TRIGGER SOURCE	External	1	The function generator does not produce an output signal until it receives a trigger on its external trigger input terminal.
	Software Trigger Function	2	The function generator does not produce an output signal until the IviFgen_SendSoftwareTrigger function is called. If you set the IVIFGEN_ATTR_TRIGGER_SOURCE property to this value, the instrument can only be triggered by calling the IviFgen_SendSoftwareTrigger function.
	Internal	3	The function generator waits until receives a trigger from its internal trigger source.
	VXI TTL0	111	The function generator waits until receives a trigger on the TTL0 line
	VXI TTL1	112	The function generator waits until receives a trigger on the TTL1 line
	VXI TTL2	113	The function generator waits until receives a trigger on the TTL2 line
	VXI TTL3	114	The function generator waits until receives a trigger on the TTL3 line
	VXI TTL4	115	The function generator waits until receives a trigger on the TTL4 line
	VXI TTL5	116	The function generator waits until receives a trigger on the TTL5 line
	VXI TTL6	117	The function generator waits until receives a trigger on the TTL6 line
	VXI TTL7	118	The function generator waits until receives a trigger on the TTL7 line

	ECL0	119	The function generator waits until receives a trigger on the ECL0 lin
	ECL1	120	The function generator waits until receives a trigger on the ECL1 lin
	PXI Star	131	The function generator waits until receives a trigger on the PXI Star
	RTSI 0	141	The function generator waits until receives a trigger on the RTSI 0 li
	RTSI 1	142	The function generator waits until receives a trigger on the RTSI 1 li
	RTSI 2	143	The function generator waits until receives a trigger on the RTSI 2 li
	RTSI 3	144	The function generator waits until receives a trigger on the RTSI 3 li
	RTSI 4	145	The function generator waits until receives a trigger on the RTSI 4 li
	RTSI 5	146	The function generator waits until receives a trigger on the RTSI 5 li
	RTSI 6	147	The function generator waits until receives a trigger on the RTSI 6 li
AM SOURCE	Internal	0	The function generator uses an internally generated waveform to modulate the output signal.
	External	1	The function generator uses a waveform from an external source modulate the output signal.
FM SOURCE	Internal	0	The function generator uses an internally generated waveform to frequency modulate the output sig
	External	1	The function generator uses a waveform from an external source modulate the output signal.

For more VI and property information:

[IviFgen VI Tree](#)

IviFgen Properties

IviFgen Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviFgen Errors	0xBFFA2001 to 0xBFFA3FFF
IviFgen Warnings	0x3FFA2001 to 0x3FFA3FFF
IVI Specific Driver Errors	0xBFFA4000 to 0xBFFA5FFF
IVI Specific Driver Warnings	0x3FFA4000 to 0x3FFA5FFF
IVI Errors	0xBFFA0000 to 0xBFFA1FFF
IVI Warnings	0x3FFA0000 to 0x3FFA1FFF
Common Instrument Driver Errors	0xBFFC0000 to 0xBFFCFFFF
Common Instrument Driver Warnings	0x3FFC0000 to 0x3FFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0x3FFFFFFF

The IviFgen class driver defines the following error codes in addition to the IVI-defined error codes.

IviFgen Error Codes

Error	Value	Message
IVIFGEN_ERROR_NO_WFMS_AVAILABLE	0xBFFA2004	The function generator's waveform memory is full.
IVIFGEN_ERROR_WFM_IN_USE	0xBFFA2008	The waveform is currently in use.
IVIFGEN_ERROR_NO_SEQS_AVAILABLE	0xBFFA2009	The function generator's sequence memory is full.
IVIFGEN_ERROR_SEQ_IN_USE	0xBFFA200D	The sequence is currently in use.
IVIFGEN_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001	The trigger source is not a software trigger.

Related Topic

[IVI Status Codes](#)

IviFgen VIs

Expand this book to view an alphabetized list of IviFgen VIs.

IviFgen Abort Generation

IviFgen Base Capability Group

This VI aborts a previously initiated signal generation. If the function generator is in the Output Generation State, this VI moves the function generator to the Configuration State. If the function generator is already in the Configuration State, the VI does nothing.



You can configure the output of the function generator regardless of whether the function generator is in the Configuration State or the Generation State. This means that you are not required to call the IviFgen Abort Generation VI prior to configuring the output of the function generator.

Many function generators constantly generate an output signal, and do not require you to abort signal generation prior to configuring the instrument. If a function generator's output cannot be aborted (that is, the function generator cannot stop generating a signal) this VI does nothing.

You are not required to call the [IviFgen Initiate Generation](#) and IviFgen Abort Generation VIs. Whether you choose to call these VIs in an application program has no impact on interchangeability. You can choose to use these VIs if you want to optimize your application for instruments that exhibit increased performance when output configuration is performed while the instrument is not generating a signal.



Note This VI does not normally check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviFgen Error-Query](#) VI at the conclusion of the sequence.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



error in (no error) The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Clear Arbitrary Memory

IviFgenArbSeq Capability Group [SEQ]

This VI removes all previously created arbitrary waveforms and sequences from the function generator's memory. This VI also invalidates all waveform and sequence handles.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

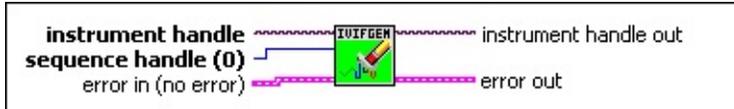
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Clear Arbitrary Sequence

IviFgenArbSeq Capability Group [SEQ]

This VI removes a previously created arbitrary sequence from the function generator's memory and invalidates the sequence's handle.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- sequence handle (0)** Pass the handle of the arbitrary sequence you want the function generator to remove. You create multiple arbitrary sequences using the [IviFgen Create Arbitrary Sequence \[SEQ\]](#) VI. The IviFgen Create Arbitrary Sequence [SEQ] VI returns a handle that you use to identify each sequence.

Defined Values:

IIVIFGEN_VAL_ALL_SEQUENCES (-1) Remove all sequences from the function generator.

Default Value: None

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

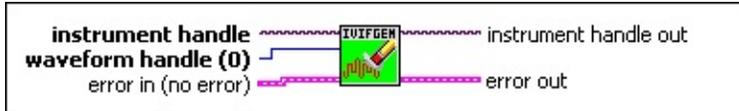
- instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Clear Arbitrary Waveform

IviFgenArbWfm Capability Group [ARB]

This VI removes a previously created arbitrary waveform from the function generator's memory and invalidates the waveform's handle.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- waveform handle (0)** Pass the handle of the arbitrary waveform you want the function generator to remove. You create multiple arbitrary waveforms using the [IviFgen Create Arbitrary Waveform \[ARB\]](#) VI. The IviFgen Create Arbitrary Waveform VI returns a handle that you use to identify each waveform.

Defined Values:

IVIFGEN_VAL_ALL_WAVEFORMS (-1)—Remove all waveforms from the function generator.

Default Value: 0

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

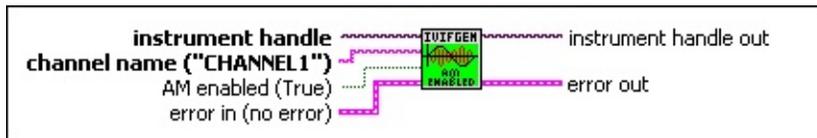
- instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure AM Enabled [AM]

IviFgenModulateAM Capability Group

This VI configures whether the function generator applies amplitude modulation to the channel.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

AM enabled (TRUE) Specify whether you want the function generator to apply amplitude modulation to the channel. The driver sets the [IviFgen AM Enabled \[AM\]](#) property to this value.

Default Value: TRUE

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



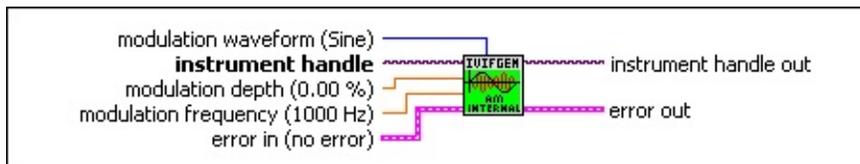
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure AM Internal

IviFgenModulateAM Capability Group [AM]

This VI configures the properties that control the function generator's internal amplitude modulation source. These properties are the modulation depth, waveform, and frequency.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

modulation depth (0.00 %) Pass the percentage of full modulation you want to apply to the carrier signal. The driver sets the [IviFgen AM Internal Depth \[AM\]](#) property to this value. At 0% depth, the output amplitude equals the carrier signal's amplitude. At 100% depth, the output amplitude equals twice the carrier signal's amplitude.

Units: Percentage (%)

Default Value: 0.0%

modulation frequency (1000 Hz) Pass the frequency at which you want the instrument to generate the modulating waveform. The driver sets the [IviFgen AM Internal Frequency \[AM\]](#) property to this value.

Units: hertz (Hz)

Default Value: 1000 Hz

modulation waveform (Sine) Pass the waveform you want to use to modulate the output signal. The driver sets the [IviFgen AM Internal Waveform \[AM\]](#) property to this value.

Defined Values:

IVIFGEN_VAL_AM_INTERNAL_SINE—Sinusoid waveform

IVIFGEN_VAL_AM_INTERNAL_SQUARE—Square waveform

IVIFGEN_VAL_AM_INTERNAL_TRIANGLE—Triangle waveform

IVIFGEN_VAL_AM_INTERNAL_RAMP_UP—Positive ramp waveform

IVIFGEN_VAL_AM_INTERNAL_RAMP_DOWN—Negative ramp waveform

IVIFGEN_VAL_AM_INTERNAL_DC—DC waveform

Default Value:

IVIFGEN_VAL_AM_INTERNAL_SINE—Sinusoid waveform



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



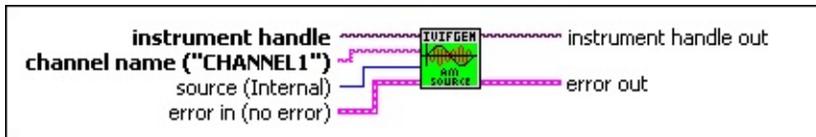
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure AM Source

IviFgenModulateAM Capability Group[xxx]

This VI configures the source of the amplitude modulating signal the function generator uses for the channel.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

source (Internal) Pass the source of the modulating signal you want to use. The driver sets the [IviFgen AM Source \[AM\]](#) property to this value.

Valid Values:

IVIFGEN_VAL_AM_INTERNAL internal modulation source

IVIFGEN_VAL_AM_EXTERNAL external modulation source

Default Value:

IVIFGEN_VAL_AM_INTERNAL internal modulation source



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



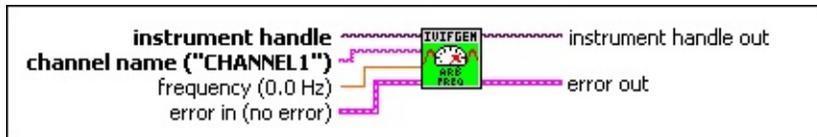
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Arb Frequency

IviFgenArbFrequency Capability Group

This VI configures the rate at which the function generator produces an entire cycle of an arbitrary waveform on a channel.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

frequency (0.0 Hz) Pass the frequency at which you want the function generator to produce one cycle of an arbitrary waveform. The driver sets the [IviFgen Arb Frequency \[AF\]](#) property to this value.

Units: hertz (Hz)

Default Value: 0.00



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



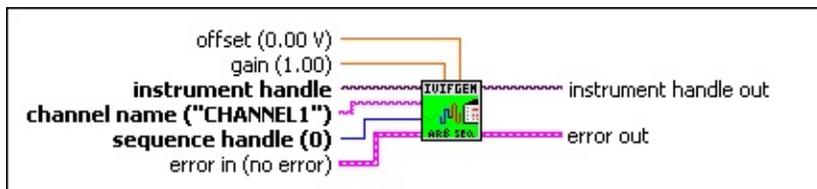
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Arbitrary Sequence

IviFgenArbSeq Capability Group [SEQ]

This VI configures the properties of the function generator that affect arbitrary sequence generation. These properties are the arbitrary sequence handle, gain, and offset.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- channel name ("CHANNEL1")** Pass the virtual channel name that you assign to the instrument in MAX.
Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.
Default Value: "CHANNEL1"
- sequence handle (0)** Pass the handle of the arbitrary sequence you want the function generator to produce. The driver sets the [IviFgen Arbitrary Sequence Handle \[SEQ\]](#) property to this value. You create an arbitrary sequence using the [IviFgen Create Arbitrary Sequence \[SEQ\]](#) VI. The IviFgen Create Arbitrary Sequence VI returns a handle that you use to identify the sequence.

Default Value: None

- gain (1.00)** Specifies the factor by which the function generator scales the arbitrary waveforms in the sequence. When you create arbitrary waveforms, you must first normalize the data points to a range of -1.00 to $+1.00$. You use this property to scale the waveforms in the sequence to other ranges. The driver sets the [IviFgen Arbitrary Waveform Gain \[ARB\]](#) property to this value. For example, to configure the output signal to range from -2.00 V to $+2.00$ V, set the gain to 2.00.

Units: None

Default Value: 1.00

- offset (0.00 V)** Specifies the value the function generator adds to the arbitrary waveforms in the sequence. When you create arbitrary waveforms, you must first normalize the data points to a range of -1.00 to $+1.00$. You use this parameter to shift the ranges of the arbitrary waveforms in the sequence. The driver sets the IviFgen [Arbitrary Waveform Offset \[ARB\]](#) property to this value. For example, to configure the output signal to range from 0.00 V to 2.00 V instead of -1.00 V to 1.00 V, set the Offset to 1.00.

Units: Volts (V)

Default Value: 0

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

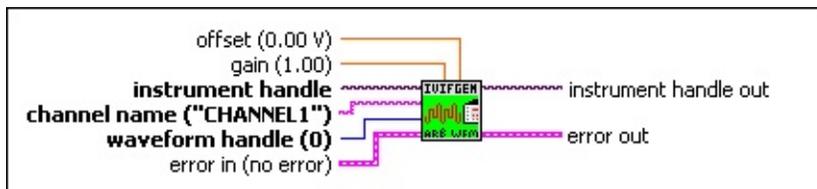
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Arbitrary Waveform

IviFgenArbWfm Capability Group [ARB]

This VI configures the properties of the function generator that affect arbitrary waveform generation. These properties are the arbitrary waveform handle, gain, and offset.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

waveform handle (0) Pass the handle of the arbitrary waveform you want the function generator to produce. The driver sets the [IviFgen Arbitrary Waveform Handle \[ARB\]](#) property to this value. You create an arbitrary waveform using the [IviFgen Create Arbitrary Waveform \[ARB\]](#) VI. The IviFgen Create Arbitrary Waveform [ARB] VI returns a handle that you use to identify the waveform.

Default Value: None

- gain (1.00)** Specifies the factor by which the function generator scales the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to a range of –1.00 to +1.00. You use this property to scale the arbitrary waveform to other ranges. The driver sets the [IviFgen Arbitrary Waveform Gain \[ARB\]](#) property to this value. For example, to configure the output signal to range from –2.00 V to +2.00 V, set the gain to 2.00.

Units: None

Default Value: 1.00

- offset (0.00 V)** Specifies the value the function generator adds to the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to a range of –1.00 to +1.00. You use this parameter to shift the arbitrary waveform's range. The driver sets the [IviFgen Arbitrary Waveform Offset \[ARB\]](#) property to this value. For example, to configure the output signal to range from 0.00 V to 2.00 V instead of –1.00 V to 1.00 V, set the Offset to 1.00.

Units: Volts (V)

Default Value: 0.0 V

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

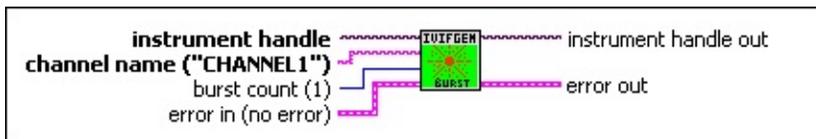
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Burst Count

IviFgenBurst Capability Group

This VI configures the burst count. When the function generator receives a trigger while in the Burst operation mode, it generates the number of waveform cycles you specify in the Burst Count control.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

burst count (1) Pass the number of waveform cycles you want the function generator to produce after it receives a trigger. The driver sets the [IviFgen Burst Count](#) property to this value.

For standard and arbitrary waveforms, a cycle is one period of the waveform.

An arbitrary sequence consists of multiple arbitrary waveforms in a sequence. Each waveform can be repeated a discrete number of times before the next waveform is produced. For arbitrary sequences, a cycle is one complete progression through the generation of all iterations of all waveforms in the sequence.

Default Value: 1



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



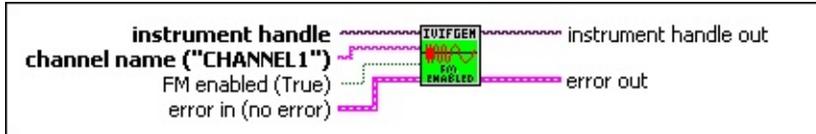
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure FM Enabled

IviFgenModulateFM Capability Group [FM]

This VI configures whether the function generator applies frequency modulation to the channel.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

FM enabled (TRUE) Specify whether you want the function generator to apply frequency modulation to the channel. The driver sets the [IviFgen FM Enabled \[FM\]](#) property to this value.

Default Value: TRUE



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



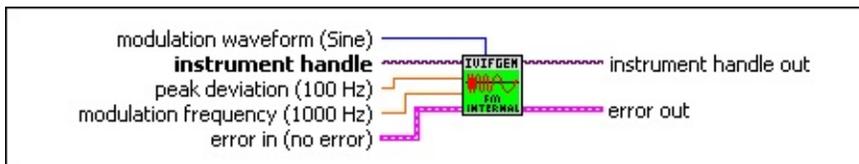
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure FM Internal

IviFgenModulateFM Capability Group [FM]

This VI configures the properties that control the function generator's internal frequency modulation source. These properties are the peak deviation, waveform, and frequency.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

peak deviation (100 Hz) Pass the maximum frequency deviation you want the function generator to apply to the carrier waveform. This deviation corresponds to the maximum amplitude level of the modulating signal. The driver sets the [IviFgen FM Internal Deviation \[FM\]](#) property to this value.

Units: hertz (Hz)

Default Value: 100.0

modulation frequency (1000 Hz) Specifies the frequency of the internal modulating signal. The driver sets the [IviFgen FM Internal Frequency \[FM\]](#) property to this value.

Units: hertz (Hz)

Default Value: 1000 Hz

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

modulation waveform (Sine) Pass the waveform you want to use to modulate the output signal. The driver sets the [IviFgen FM](#)

[Internal Waveform \[FM\]](#) property to this value.

Defined Values:

IVIFGEN_VAL_FM_INTERNAL_SINE Sinusoid waveform

IVIFGEN_VAL_FM_INTERNAL_SQUARE Square waveform

IVIFGEN_VAL_FM_INTERNAL_TRIANGLE Triangle waveform

IVIFGEN_VAL_FM_INTERNAL_RAMP_UP Positive ramp waveform

IVIFGEN_VAL_FM_INTERNAL_RAMP_DOWN Negative ramp waveform

Default Value:

IVIFGEN_VAL_FM_INTERNAL_SINE Sinusoid waveform



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



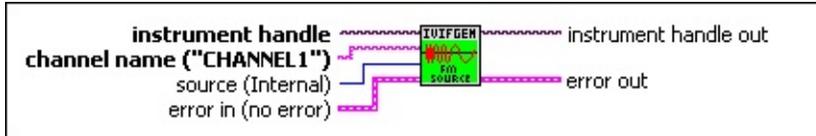
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure FM Source

IviFgenModulateFM Capability Group

This VI configures the source of the frequency modulating signal the function generator uses for the channel.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

source (Internal) Pass the source of the modulating signal you want to use. The driver sets the [IviFgen FM Source \[FM\]](#) property to this value.

Valid Values:

IVIFGEN_VAL_FM_INTERNAL internal modulation source

IVIFGEN_VAL_FM_EXTERNAL external modulation source

Default:

IVIFGEN_VAL_FM_INTERNAL internal modulation source

 **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



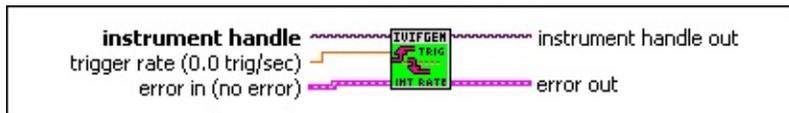
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Internal Trigger Rate

IviFgenInternalTrigger Capability Group

This VI configures the rate at which the function generator's internal trigger source generates trigger signals.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

Cluster **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

trigger rate Specify the rate at which you want the function generator's internal trigger rate to generate trigger signals. The driver sets the [IviFgen Internal Trigger Rate \[IT\]](#) to this value.

I/O **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

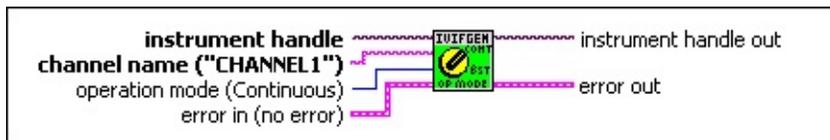
Cluster **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Operation Mode

IviFgen Base Capability Group

This VI configures the operation mode of the function generator. The operation mode determines how the function generator produces output on a channel. For example, you can select to generate output continuously or to generate a discrete number of waveform cycles based on a trigger event.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- operation mode (Continuous)** Pass the operation mode you want the function generator to use. The driver sets the [IviFgen Operation Mode](#) property to this value.

Defined Values:

IVIFGEN_VAL_OPERATE_CONTINUOUS—continuous generation

IVIFGEN_VAL_OPERATE_BURST—burst generation

Default Value:

IVIFGEN_VAL_OPERATE_CONTINUOUS—continuous generation

-  **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

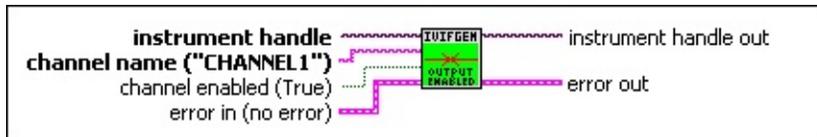
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Output Enabled

IviFgen Base Capability Group

This VI configures whether the signal the function generator produces appears at the channel's output connector.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- channel name ("CHANNEL1")** If the property is channel-based, this parameter specifies the name of the channel on which to set the value of the property. If the property is not channel-based, then pass VI_NULL or an empty string.

Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

channel enabled (True) Specify whether you want the signal the function generator produces to appear at the output terminal. The driver sets the [IviFgen Output Enabled](#) property to this value.

Default Value: TRUE

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

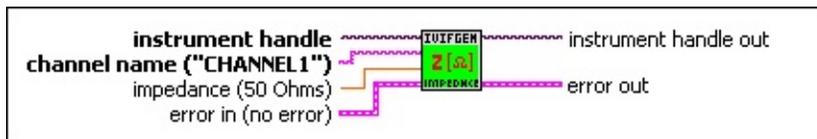
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Output Impedance

IviFgen Base Capability Group

This VI configures the output impedance for the channel you specify.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

impedance (50 ohms) Pass the impedance value you want the function generator to use. The driver sets the [IviFgen Output Impedance](#) property to this value.

A value of 0.0 indicates that the function generator is connected to a high-impedance load.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



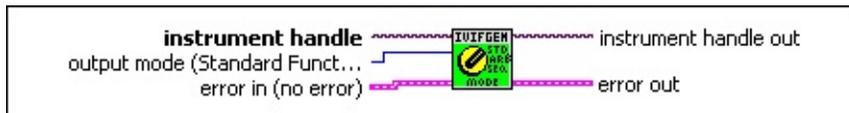
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Output Mode

IviFgen Base Capability Group

This VI configures the output mode of the function generator. The output mode determines the kind of waveform the function generator produces. For example, you can select to output a standard waveform, an arbitrary waveform, or a sequence of arbitrary waveforms.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- output mode (Standard Function)** Pass the output mode you want the function generator to use. The driver sets the IviFgen Output Mode property to this value. The value you specify determines which VIs and properties you use to configure the waveform the function generator produces.

When you set this parameter to `IVIFGEN_VAL_OUTPUT_FUNC`, use the following VI to configure the waveform:

[IviFgen Configure Standard Waveform \[STD\]](#)

In place of using this VI, you can set the following properties:

[IviFgen Waveform \[STD\]](#)

[IviFgen Func Amplitude \[STD\]](#)

[IviFgen DC Offset \[STD\]](#)

[IviFgen Frequency \[STD\]](#)

[IviFgen Start Phase \[STD\]](#)

[IviFgen Duty Cycle High \[STD\]](#)

When you set this parameter to `IVIFGEN_VAL_OUTPUT_ARB`, use the following VIs to configure the waveform:

[IviFgen Query Arb Waveform Capabilities \[ARB\]](#)

[IviFgen Create Arbitrary Waveform \[ARB\]](#)

[IviFgen Clear Arbitrary Waveform \[ARB\]](#)

[IviFgen Configure Sample Rate \[ARB\]](#)

[IviFgen Configure Arbitrary Waveform \[ARB\]](#)

[IviFgen Configure Arb Frequency \[AF\]](#)

Instead of using the [IviFgen Configure Sample Rate \[ARB\]](#) and [IviFgen Configure Arbitrary Waveform \[ARB\]](#) VIs, you can set the following properties:

[IviFgen Arbitrary Waveform Handle \[ARB\]](#)

[IviFgen Arbitrary Waveform Gain \[ARB\]](#)

[IviFgen Arbitrary Waveform Offset \[ARB\]](#)

[IviFgen Arb Sample Rate \[ARB\]](#)

[IviFgen Arb Frequency \[AF\]](#)

When you set this parameter to IVIFGEN_VAL_OUTPUT_SEQ, use the following VIs to configure the sequence:

[IviFgen Query Arb Waveform Capabilities \[ARB\]](#)

[IviFgen Create Arbitrary Waveform \[ARB\]](#)

[IviFgen Clear Arbitrary Waveform \[ARB\]](#)

[IviFgen Query Arb Sequence Capabilities \[SEQ\]](#)

[IviFgen Clear Arbitrary Sequence \[SEQ\]](#)

[IviFgen Clear Arbitrary Memory \[SEQ\]](#)

[IviFgen Configure Sample Rate \[ARB\]](#)

[IviFgen Configure Arbitrary Sequence \[SEQ\]](#)

Instead of using the [IviFgen Configure Sample Rate \[ARB\]](#) and [IviFgen Configure Arbitrary Sequence \[SEQ\]](#) VIs, you can set the following properties:

IviFgen [Arbitrary Sequence Handle \[SEQ\]](#)

IviFgen [Arbitrary Waveform Gain \[ARB\]](#)

IviFgen [Arbitrary Waveform Offset \[ARB\]](#)

IviFgen [Arb Sample Rate \[ARB\]](#)

Defined Values:

IVIFGEN_VAL_OUTPUT_FUNC Standard Function Output

IVIFGEN_VAL_OUTPUT_ARB Arbitrary Waveform Output

IVIFGEN_VAL_OUTPUT_SEQ Arbitrary Sequence output

Default Value:

IVIFGEN_VAL_OUTPUT_FUNC Standard Function Output

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

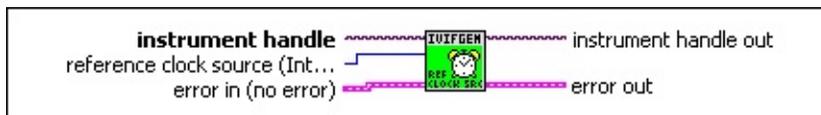
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Ref Clock Source

IviFgen Base Capability Group

This VI configures the function generator's reference clock source. The function generator uses the reference clock to derive frequencies and sample rates when generating output.



IVI **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

ESI **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

reference clock source (Internal) Pass the reference clock source you want the function generator to use. The driver sets the [IviFgen Reference Clock Source](#) property to this value.

The function generator derives the frequencies and sample rates that it uses to generate waveforms from the source you specify. For example, when you set this parameter to `IVIFGEN_VAL_REF_CLOCK_EXTERNAL`, the function generator uses the signal it receives at its external clock terminal as its reference clock.

Defined Values:

`IVIFGEN_VAL_REF_CLOCK_INTERNAL`—Internal clock source

`IVIFGEN_VAL_REF_CLOCK_EXTERNAL`—External clock source

`IVIFGEN_VAL_REF_CLOCK_RTSM_CLOCK`—RTSM clock source

Default:

IVIFGEN_VAL_REF_CLOCK_INTERNAL—Internal clock source



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



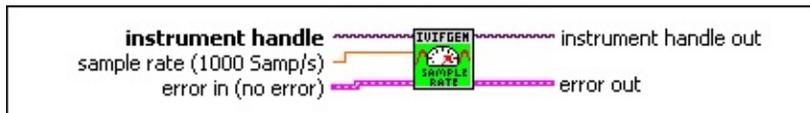
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Sample Rate

IviFgenArbWfm Capability Group [ARB]

This VI configures the rate at which the function generator produces the points that make up arbitrary waveforms. When you configure the function generator to produce an arbitrary sequence, this is the sample rate for all arbitrary waveforms in the sequence.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

sample rate (1000 Samp/s) Pass the sample rate at which you want the function generator to output arbitrary waveforms. The driver sets the [IviFgen Arb Sample Rate \[ARB\]](#) property to this value.

Units: Samples-per-Second (Samp/s)

Default Value: 1000

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

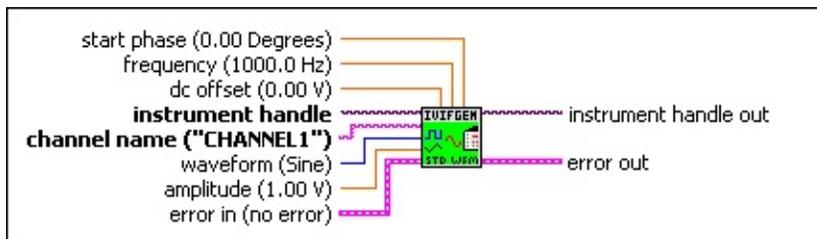
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Standard Waveform

IviFgenStdFunc Capability Group [STD]

This VI configures the properties of the function generator that affect standard waveform generation. These properties are the waveform, amplitude, DC offset, frequency, and start phase.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

waveform (Sine) Pass the standard waveform that you want the function generator to produce. The driver sets the [IviFgen Waveform \[STD\]](#) property to this value.

Valid Values:

IIVIFGEN_VAL_WFM_SINE—Sinusoid waveform

IVIFGEN_VAL_WFM_SQUARE—Square waveform

IVIFGEN_VAL_WFM_TRIANGLE—Triangle waveform

IVIFGEN_VAL_WFM_RAMP_UP—Positive ramp waveform

IVIFGEN_VAL_WFM_RAMP_DOWN—Negative ramp waveform

IVIFGEN_VAL_WFM_DC—Constant voltage

Default Value:

IVIFGEN_VAL_WFM_SINE—Sinusoid waveform

- amplitude (1.0 V)** Pass the amplitude of the standard waveform that you want the function generator to produce. This value is the amplitude at the output terminal. The driver sets the [lviFgen Func Amplitude \[STD\]](#) property to this value. For example, to produce a waveform ranging from –5.00 V to +5.00 V, set the amplitude to 10.00 V.

Units: Volts peak-to-peak (Vpp)

Default Value: 1.0 V

-  **Note** This parameter does not affect function generator behavior when you set the **Waveform** parameter to IVIFGEN_VAL_WFM_DC.

- dc offset (0.00 V)** Pass the DC offset of the standard waveform that you want the function generator to produce. This value is the offset at the output terminal. The value is the offset from ground to the center of the waveform you specify with the **Waveform** parameter. The driver sets the [lviFgen DC Offset \[STD\]](#) property to this value. For example, to configure a waveform with an amplitude of 10.00 volts to range from 0.00 V to +10.00 V, set the DC Offset to 5.00 V.

Units: volts (V)

Default Value: 0.0

- frequency (0.0 Hz)** Pass the frequency of the standard waveform that you want the function generator to produce. The driver sets the [lviFgen Frequency \[STD\]](#) property to this value.

Units: hertz (Hz)

Default Value: 1000.0 Hz



Note This parameter does not affect function generator behavior when you set the **Waveform** parameter to `IVIFGEN_VAL_WFM_DC`.



start phase (0.00 Degrees) Pass the horizontal offset of the standard waveform you want the function generator to produce. You specify this property in degrees of one waveform cycle. The driver sets the [IviFgen Start Phase \[STD\]](#) property to this value. A start phase of 180 degrees means output generation begins half way through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle, which is identical to a start phase of 0 degrees.

Units: Degrees of one cycle

Default Value: 0.0



Note This parameter does not affect function generator behavior when you set the **Waveform** parameter to `IVIFGEN_VAL_WFM_DC`.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.



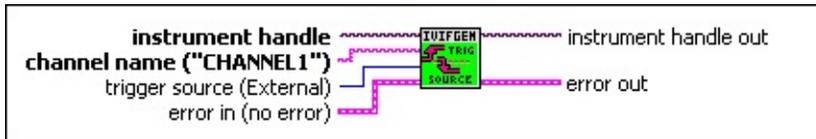
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Configure Trigger Source

IviFgenTrigger Capability Group [TRG]

This VI configures the trigger source for a channel.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

trigger source (External) Specify the trigger source you want to use. The driver sets the [IviFgen Trigger Source \[TRG\]](#) property to this value.

After you call the [IviFgen InitiateGeneration](#) VI, the function generator waits for the trigger you specify in this parameter. After it

receives a trigger, the function generator produces output based on the value of the [IviFgen Operation Mode](#) property.

Defined Values:

IVIFGEN_VAL_INTERNAL_TRIGGER—Internal

IVIFGEN_VAL_EXTERNAL—External

IVIFGEN_VAL_SOFTWARE_TRIG—Software Trigger Function

IVIFGEN_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVIFGEN_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVIFGEN_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVIFGEN_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVIFGEN_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVIFGEN_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVIFGEN_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVIFGEN_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVIFGEN_VAL_ECL0—VXI ECL0

IVIFGEN_VAL_ECL1—VXI ECL1

IVIFGEN_VAL_PXI_STAR—PXI Star

IVIFGEN_VAL_RTSL_0—RTSL line 0

IVIFGEN_VAL_RTSL_1—RTSL line 1

IVIFGEN_VAL_RTSL_2—RTSL line 2

IVIFGEN_VAL_RTSL_3—RTSL line 3

IVIFGEN_VAL_RTSL_4—RTSL line 4

IVIFGEN_VAL_RTSL_5—RTSL line 5

IVIFGEN_VAL_RTSL_6—RTSL line 6

Default Value:

IVIFGEN_VAL_EXTERNAL—External

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

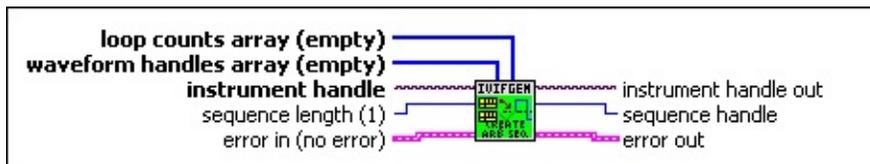
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Create Arbitrary Sequence

IviFgenArbSeq Capability Group [SEQ]

An arbitrary sequence consists of multiple waveforms. For each waveform, you specify the number of times the function generator produces the waveform before proceeding to the next waveform. The number of times to repeat a specific waveform is called the loop count.

This VI creates an arbitrary sequence from an array of waveform handles and an array of corresponding loop counts. The VI returns a handle that identifies the sequence. You pass this handle to the [IviFgen Configure Arbitrary Sequence \[SEQ\]](#) VI to specify what arbitrary sequence you want the function generator to produce.



- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.
- sequence length (1)** Pass the number of waveforms in the new arbitrary sequence you want to create. The value you pass must be between the minimum and maximum sequence lengths the function generator allows. You can obtain the minimum and maximum sequence lengths from the **Minimum Sequence Length** and **Maximum Sequence Length** parameters of the [IviFgen Query Arb Sequence Capabilities \[SEQ\]](#) VI.

Default Value: 1

- waveform handles array (empty)** Pass the array of waveform handles from which you want to create a new arbitrary sequence. The array must have at least as many elements as the value you specify in the **Sequence Length** parameter. Each Waveform Handles array element has a corresponding Loop Counts array element that indicates how many times that waveform is repeated.

You obtain waveform handles when you create arbitrary waveforms with the [IviFgen Create Arbitrary Waveform \[ARB\]](#) VI.

Default Value: None

- loop counts array (empty)** Pass the array of loop counts you want to use to create a new arbitrary sequence. The array must have at least as many elements as the value you specify in the **Sequence Length** parameter. Each Loop Counts array element corresponds to a Waveform Handles array element and indicates how many times to repeat that waveform. Each element of the Loop Count array must be less than or equal to the maximum number of loop counts the function generator allows. You can obtain the maximum loop count from the **Maximum Loop Count** parameter of the [IviFgen Query Arb Sequence Capabilities \[SEQ\]](#) VI.

Default Value: None

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

- sequence handle** Returns the handle that identifies the new arbitrary sequence. You pass this handle to the [IviFgen Configure Arbitrary Sequence \[SEQ\]](#) VI to generate the arbitrary sequence.

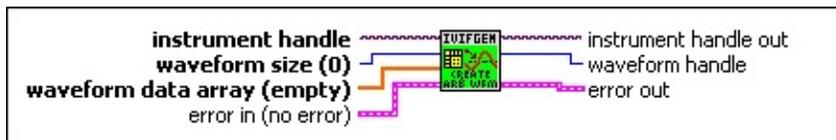
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Create Arbitrary Waveform

IviFgenArbWfm Capability Group [ARB]

This VI creates an arbitrary waveform and returns a handle that identifies that waveform. You pass this handle to the [IviFgen Configure Arbitrary Waveform \[ARB\]](#) VI to produce that waveform. You also use the handles this VI returns to specify a sequence of arbitrary waveforms with the [IviFgen Create Arbitrary Sequence \[SEQ\]](#) VI.



instrument handle The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

waveform size (0) Pass the size of the arbitrary waveform you want create.

The size must meet the following restrictions:

- The size is less than or equal to the maximum waveform size the instrument allows.
- The size is greater than or equal to the minimum waveform size the instrument allows.
- The size is an integer multiple of the instrument's waveform quantum.

You can obtain these values from the **Maximum Waveform Size**, **Minimum Waveform Size**, and **Waveform Quantum** parameters of the [IviFgen Query Arb Waveform Capabilities \[ARB\]](#) VI.

Default Value: None

waveform data array (empty) Specify the array of data you want to use for the new arbitrary waveform. The array must have at least as many elements as the value you specify in the Waveform Size parameter.

You must normalize the data points in the array to be between – 1.00 and +1.00.

Default Value: None

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

waveform handle Returns the handle that identifies the new arbitrary waveform. You pass this handle to the [IviFgen Configure Arbitrary Waveform \[ARB\]](#) VI to generate the arbitrary waveform. You pass an array of these handles to the [IviFgen Create Arbitrary Sequence \[SEQ\]](#) VI to create an arbitrary sequence.

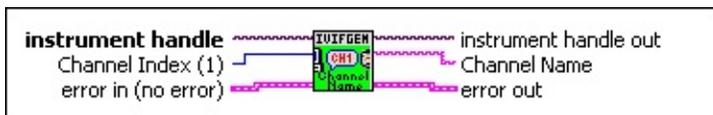
 **[error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Get Channel Name

IviFgen Base Capability Group

This VI returns the specific driver defined channel name that corresponds to the one-based index that you specify. If the value you pass for the **Index** parameter is less than one or greater than the value of the Channel Count property, the VI returns an empty string in the **Name** parameter and returns an error.

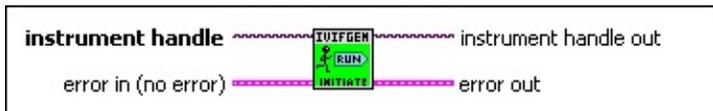


- instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI. The handle identifies a particular instrument session.
- index (1)** A one-based index that defines which name to return. The index must be less than or equal to the [ClassPrefix Channel Count](#) property.
- instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
- output channel name** Returns the name of the Channel specified by the **index** parameter.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Initiate Generation

IviFgen Base Capability Group

This VI initiates signal generation. If the function generator is in the Configuration state, this VI moves the function generator to the Output Generation state. If the function generator is already in the Output Generation state, this VI does nothing. [Details](#).



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Initiate Generation Details

The instrument is in the Generation state after you call the [IviFgen Initialize](#), [IviFgen Initialize With Options](#), or [IviFgen Reset](#) VIs. You can configure the output of the function generator regardless of whether function generator is in the Configuration State or the Generation State. This means that you are required to call the IviFgen Initiate Generation VI only if you abort signal generation by calling the [IviFgen Abort Generation](#)

VI.

Many function generators constantly generate an output signal, and do not require you to initiate signal generation. If a function generator is always outputting the currently configured signal, this VI does nothing.

You are not required to call the IviFgen Initiate Generation and [IviFgen Abort Generation](#) VIs. Whether you choose to call these VIs in an application program has no impact on interchangeability. You can choose to use these VIs if you want to optimize your application for instruments that exhibit increased performance when output configuration is performed while the instrument is not generating a signal.

This VI disables extensions that you have not set under the following conditions:

1. If you have not set the value of any property in the IviFgenModulateAM extension on a channel, this VI sets the IviFgen»Amplitude Modulation»[AM Enabled \[AM\]](#) property to FALSE for that channel.
2. If you have not set the value of any property in the IviFgenModulateFM extension on a channel, this VI sets the IviFgen»Frequency Modulation»[FM Enabled \[FM\]](#) property to FALSE for that channel.



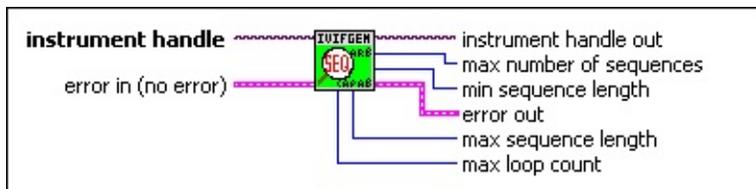
Notes

1. This VI does not normally check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver functions. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the IviFgen Error-Query VI at the conclusion of the sequence.
2. This VI performs interchangeability checking when the [IviFgen Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternatively, you may use the [IviFgen Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviFgen Interchange Check](#) property.

IviFgen Query Arb Sequence Capabilities

IviFgenArbSeq Capability Group

This VI returns the properties of the function generator that are related to creating arbitrary sequences. These properties are the maximum number of sequences, minimum sequence length, maximum sequence length, and maximum loop count.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

max number of sequences Returns the maximum number of arbitrary waveform sequences that the function generator allows. The driver obtains this value from the [IviFgen Max Number of Sequences](#) property.

min sequence length Returns the minimum number of arbitrary waveforms the function generator allows in a sequence. The driver obtains this value from the IviFgen [Min Sequence Length](#) property.

max sequence length Returns the maximum number of arbitrary waveforms the function generator allows in a sequence. The driver obtains this value from the [IviFgen Max Sequence Length](#) property.

max loop count Returns the maximum number of times the function generator can repeat an arbitrary waveform in a sequence. The driver obtains this value from the [IviEgen Max Loop Count](#) property.

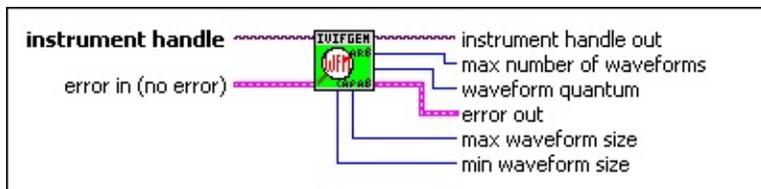
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Query Arb Waveform Capabilities

IviFgenArbWfm Capability Group [SEQ]

This VI returns the properties of the function generator that are related to creating arbitrary waveforms. These properties are the maximum number of waveforms, waveform quantum, minimum waveform size, and maximum waveform size.



I/O **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

max number of waveforms Returns the maximum number of arbitrary waveforms that the function generator allows. The driver obtains this value from the [IviFgen Max Number of Waveforms \[ARB\]](#) property.

waveform quantum The size (such as, number of points) of each waveform must be a multiple of a constant quantum value. This parameter obtains the quantum value the function generator uses. The driver returns this value from the [IviFgen Waveform Quantum \[ARB\]](#) property. For example, when this property returns a value of 8, all waveform sizes must be a multiple of 8.

- min waveform size** Returns the minimum number of points the function generator allows in a waveform. The driver obtains this value from the [IviFgen Min Waveform Size \[ARB\]](#) property.
- max waveform size** Returns the maximum number of points the function generator allows in a waveform. The driver obtains this value from the [IviFgen Max Waveform Size \[ARB\]](#) property.
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

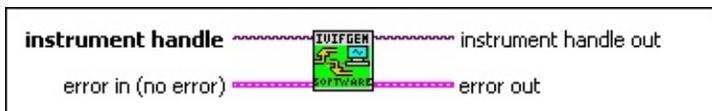
IviFgen Send Software Trigger

IviFgenSoftwareTrigger Capability Group [SWT]

This VI sends a command to trigger the instrument. You must satisfy certain conditions before you can successfully call this VI, The [IviFgen Trigger Source](#) is set to IVIDMM_VAL_SOFTWARE_TRIG.

If this condition is not true, this VI does not send the software trigger and returns the error IVIFGEN_ERROR_TRIGGER_NOT_SOFTWARE.

This VI does not check the instrument status. Typically, the end-user calls this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. The end-user uses the low-level VIs to optimize one or more aspects of interaction with the instrument. To check the instrument status, call the [IviFgen Error-Query](#) VI at the conclusion of the sequence.



 **instrument handle** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviFgen Initialize](#) or [IviFgen Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviFgen Properties

Expand this book to view an alphabetized list of IviFgen properties.

IviFgen AM Enabled [AM]

IviFgenModulateAM Property [AM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure AM Enabled

Description

This channel-based property specifies whether the function generator applies amplitude modulation to the output signal.

Property Node Path

IviFgen»Amplitude Modulation»AM Enabled [AM]

IviFgen AM Internal Depth [AM]

IviFgenModulateAM Property [AM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure AM Internal

Description

Specifies the extent of modulation the function generator applies to the carrier signal. The units are a percentage of full modulation. At 0% depth, the output amplitude equals the carrier signal's amplitude. At 100% depth, the output amplitude equals twice the carrier signal's amplitude.

This property affects function generator behavior only when the [IviFgen AM Source \[AM\]](#) property is set to IVIFGEN_VAL_AM_INTERNAL.

Property Node Path

IviFgen»Amplitude Modulation»AM Internal Depth [AM]

IviFgen AM Internal Frequency

IviFgenModulateAM Property [AM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure AM Internal

Description

Specifies the frequency of the standard waveform that the function generator uses to modulate the output signal. The units are hertz (Hz).

This property affects function generator behavior only when the [IviFgen AM Source \[AM\]](#) property is set to IVIFGEN_VAL_AM_INTERNAL.

Property Node Path

IviFgen»Amplitude Modulation»AM Internal Frequency

IviFgen AM Internal Waveform [AM]

IviFgenModulateAM Property [AM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure AM Internal

Description

Specifies the standard waveform type that the function generator uses to modulate the output signal.

This property affects function generator behavior only when the [IviFgen AM Source \[AM\]](#) property is set to IVIFGEN_VAL_AM_INTERNAL.

Property Node Path

IviFgen»Amplitude Modulation»AM Internal Waveform [AM]

Defined Values

[IVIFGEN_VAL_AM_INTERNAL_SINE](#)

[IVIFGEN_VAL_AM_INTERNAL_SQUARE](#)

[IVIFGEN_VAL_AM_INTERNAL_TRIANGLE](#)

[IVIFGEN_VAL_AM_INTERNAL_RAMP_UP](#)

[IVIFGEN_VAL_AM_INTERNAL_RAMP_DOWN](#)

IviFgen AM Source [AM]

IviFgenModulateAM Property [AM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure AM Source

Description

This channel-based property specifies the signal that the function generator uses to modulate the output signal.

Property Node Path

IviFgen»Amplitude Modulation»AM Source [AM]

Defined Values

[IVIFGEN_VAL_AM_INTERNAL](#)

[IVIFGEN_VAL_AM_EXTERNAL](#)

IviFgen Arb Frequency [AF]

IviFgenArbFrequency Property [AF]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Arb Frequency

Description

Specifies the rate at which the function generator outputs an entire arbitrary waveform. The units are arbitrary waveforms per second.

You use this property when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_ARB.

Property Node Path

IviFgen»Arbitrary Waveform Output»Arbitrary Waveform Frequency»Arb Frequency [AF]

IviFgen Arb Sample Rate [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure Sample Rate [ARB]

Description

Specifies the rate at which the function generator outputs the points in arbitrary waveforms. The units are samples per second.

You use this property when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_ARB or IVIFGEN_VAL_OUTPUT_SEQ.

Property Node Path

IviFgen»Arbitrary Waveform Output»Arb Sample Rate [ARB]

IviFgen Arbitrary Sequence Handle [SEQ]

IviFgenArbSeq Property [SEQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Arbitrary Sequence

Description

This channel-based property identifies which sequence the function generator produces. You can create multiple sequences using the [IviFgen Create Arbitrary Sequence \[SEQ\]](#) VI. The Create Arbitrary Sequence [SEQ] VI returns a handle that you use to identify the particular sequence. To configure the function generator to produce a particular sequence, you set this property to the sequence's handle.

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_SEQ.

Property Node Path

IviFgen»Arbitrary Sequence Output»Arbitrary Sequence Handle [SEQ]

IviFgen Arbitrary Waveform Gain [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Arbitrary Waveform [ARB]

Description

This channel-based property specifies the factor by which the function generator scales the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to the range -1.0 to $+1.0$. You use this property to scale the arbitrary waveform to other ranges. For example, when you set this property to 2.0 , the output signal ranges from -2.0 to $+2.0$ V.

You use this property when the [IviFgen Output Mode](#) property is set to `IVIFGEN_VAL_OUTPUT_ARB` or `IVIFGEN_VAL_OUTPUT_SEQ`.

Property Node Path

IviFgen»Arbitrary Waveform Output»Arbitrary Waveform Gain [ARB]

IviFgen Arbitrary Waveform Handle [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Arbitrary Waveform [ARB]

Description

This channel-based property identifies which arbitrary waveform the function generator produces. You can create multiple arbitrary waveforms using the [IviFgen Create Arbitrary Waveform VI](#). The Create Arbitrary Waveform [ARB] VI returns a handle that you use to identify the particular waveform. To configure the function generator to produce a particular waveform, you set this property to the waveform's handle.

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_ARB.

Property Node Path

IviFgen»Arbitrary Waveform Output»Arbitrary Waveform Handle [ARB]

IviFgen Arbitrary Waveform Offset [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Arbitrary Waveform [ARB]

Description

This channel-based property specifies the value the function generator adds to the arbitrary waveform data. When you create arbitrary waveforms, you must first normalize the data points to the range -1.0 to $+1.0$. You use this property to shift the arbitrary waveform's range. The units are volts (V). For example, when you set this property to 1.0 , the output signal ranges from 2.0 V to 0.0 V.

You use this property when the [IviFgen Output Mode](#) property is set to `IVIFGEN_VAL_OUTPUT_ARB` or `IVIFGEN_VAL_OUTPUT_SEQ`.

Property Node Path

IviFgen»Arbitrary Waveform Output»Arbitrary Waveform Offset [ARB]

IviFgen Burst Count [BST]

IviFgenBurst Property [BST]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Burst Count

Description

This channel-based property specifies the number of waveform cycles that the function generator produces after it receives a trigger.

This property affects function generator behavior only when the

[IviFgen Operation Mode](#) property is set to IVIFGEN_VAL_OPERATE_BURST.

Property Node Path

IviFgen»Burst Operation»Burst Count [BST]

IviFgen Channel Count

IviFgen Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of channels available on the instrument.

Property Node Path

IviFgen»Inherent IVI Settings»Specific Driver Capabilities»Channel Count

IviFgen DC Offset [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Standard Waveform

Description

This channel-based property specifies the DC offset of the standard waveform the function generator produces. This value is the offset at the output terminal. The units are volts (V). The value is the offset from ground to the center of the waveform you specify with the [IviFgen Waveform \[STD\]](#) property. For example, a standard waveform ranging from +5.0 V to 0.0 V has a DC offset of 2.5 V.

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC.

Property Node Path

IviFgen»Standard Function Output»DC Offset [STD]

IviFgen Duty Cycle High [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	None

Description

This channel-based property specifies the length of time the output voltage level remains high in a square waveform. You specify this property as a percentage of one waveform cycle.

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC and the [IviFgen Waveform \[STD\] property to IVIFGEN_VAL_WFM_SQUARE](#).

Property Node Path

IviFgen»Standard Function Output»Duty Cycle High [STD]

IviFgen FM Enabled [FM]

IviFgenModulateFM Property [FM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure FM Enabled

Description

Specifies whether the function generator applies frequency modulation to the output signal.

Property Node Path

IviFgen»Frequency Modulation»FM Enabled [FM]

IviFgen FM Internal Frequency [FM]

IviFgenModulateFM Property [FM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure FM Internal

Description

Specifies the frequency of the standard waveform that the function generator uses to modulate the output signal. The units are hertz (Hz).

This property affects function generator behavior only when the [IviFgen FM Source \[FM\]](#) property is set to IVIFGEN_VAL_FM_INTERNAL.

Property Node Path

IviFgen»Frequency Modulation»FM Internal Frequency [FM]

IviFgen FM Internal Waveform

IviFgenModulateFM Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure FM Internal

Description

Specifies the standard waveform type that the function generator uses to modulate the output signal.

This property affects function generator behavior only when the [IviFgen FM Source](#) property is set to IVIFGEN_VAL_FM_INTERNAL.

Property Node Path

IviFgen»Frequency Modulation»FM Internal Waveform

Defined Values

[IVIFGEN_VAL_FM_INTERNAL_SINE](#)

[IVIFGEN_VAL_FM_INTERNAL_SQUARE](#)

[IVIFGEN_VAL_FM_INTERNAL_TRIANGLE](#)

[IVIFGEN_VAL_FM_INTERNAL_RAMP_UP](#)

[IVIFGEN_VAL_FM_INTERNAL_RAMP_DOWN](#)

IviFgen FM Internal Deviation [FM]

IviFgenModulateFM Property [FM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure FM Internal

Description

Specifies the maximum frequency deviation the modulating waveform applies to the carrier waveform. This deviation corresponds to the maximum amplitude level of the modulating signal. The units are hertz (Hz).

This property affects function generator behavior only when the [IviFgen FM Source \[FM\]](#) property is set to IVIFGEN_VAL_FM_INTERNAL.

Property Node Path

IviFgen»Frequency Modulation»FM Internal Deviation [FM]

IviFgen FM Source [FM]

IviFgenModulateFM Property [FM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure FM Source

Description

Specifies the signal the function generator uses to modulate the output signal.

Property Node Path

IviFgen»Frequency Modulation»FM Source [FM]

Defined Values

[IVIFGEN_VAL_FM_INTERNAL](#)

[IVIFGEN_VAL_FM_EXTERNAL](#)

IviFgen Frequency [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Standard Waveform

Description

This channel-based property specifies the frequency of the standard waveform the function generator produces. The units are hertz (Hz). You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC.

This property does not affect function generator behavior when you set the [IviFgen Waveform \[STD\]](#) property to IVIFGEN_VAL_WFM_DC.

Property Node Path

IviFgen»Standard Function Output»Frequency [STD]

IviFgen Func Amplitude [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Standard Waveform

Description

This channel-based property specifies the amplitude of the standard waveform the function generator produces. This value is the amplitude at the output terminal. The units are volts peak-to-peak (Vpp). For example, to produce a waveform ranging from -5.0 to +5.0 V, you set this value to 10.0 V.

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC. This property does not affect function generator behavior when you set the [IviFgen Waveform \[STD\]](#) property to IVIFGEN_VAL_WFM_DC.

Property Node Path

IviFgen»Standard Function Output»Func Amplitude [STD]

IviFgen Internal Trigger Rate [IT]

IviFgenInternalTrigger Property [ITRG]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Internal Trigger Rate

Description

This property specifies the rate at which the function generator's internal trigger source produces a trigger, in triggers per second.

This property affects function generator behavior only when the [IviFgen Trigger Source \[TRG\]](#) property is set to IVIFGEN_VAL_INTERNAL_TRIGGER on a channel.

Property Node Path

IviFgen»Triggering»Internal Trigger»Internal Trigger Rate [IT]

IviFgen Max Loop Count [SEQ]

IviFgenArbSeq Property [SEQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Sequence Capabilities

Description

Returns the maximum number of times the function generator can repeat a waveform in a sequence. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Sequence Output»Max Loop Count

IviFgen Max Number of Sequences [SEQ]

IviFgenArbSeq Property [SEQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Sequence Capabilities

Description

Returns the maximum number of arbitrary sequences the function generator allows. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Sequence Output»Max Number of Sequences [SEQ]

IviFgen Max Number of Waveforms [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Waveform Capabilities

Description

Returns the maximum number of arbitrary waveforms that the function generator allows. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Waveform Output»Max Number of Waveforms [ARB]

IviFgen Max Sequence Length [SEQ]

IviFgenArbSeq Property [SEQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Sequence Capabilities

Description

Returns the maximum number of arbitrary waveforms the function generator allows in a sequence. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Sequence Output»Max Sequence Length [SEQ]

IviFgen Max Waveform Size [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Waveform Capabilities

Description

Returns the maximum number of points the function generator allows in an arbitrary waveform. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Waveform Output»Max Waveform Size [ARB]

IviFgen Min Sequence Length [SEQ]

IviFgenArbSeq Property [SEQ]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Sequence Capabilities

Description

Returns the minimum number of arbitrary waveforms the function generator allows in a sequence. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Sequence Output»Min Sequence Length [SEQ]

IviFgen Min Waveform Size [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Waveform Capabilities

Description

Returns the minimum number of points the function generator allows in an arbitrary waveform. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Waveform Output»Min Waveform Size [ARB]

IviFgen Operation Mode

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Operation Mode

Description

This property specifies how the function generator produces waveforms. For example, you can configure the instrument to generate output continuously, or to generate a discrete number of waveform cycles based on a trigger event.

Property Node Path

IviFgen»Basic Operation»Operation Mode

Defined Values

[IVIFGEN_VAL_OPERATE_CONTINUOUS](#)

[IVIFGEN_VAL_OPERATE_BURST](#)

IviFgen Output Enabled

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Output Enabled

Description

This channel-based property specifies whether the signal the function generator produces appears at the output connector.

Property Node Path

IviFgen»Basic Operation»Output Enabled

IviFgen Output Impedance

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Output Impedance

Description

This channel-based property specifies the function generator's output impedance at the output connector.

Property Node Path

IviFgen»Basic Operation»Output Impedance

IviFgen Output Mode

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure Operation Mode

Description

Determines the kind of waveform the function generator produces. The value you specify determines which VIs and properties you use to configure the waveform the function generator produces.

When you set this property to `IVIFGEN_VAL_OUTPUT_FUNC`, use the following properties and VIs to configure standard waveforms:

[IviFgen Waveform \[STD\]](#)

[IviFgen Func Amplitude \[STD\]](#)

[IviFgen DC Offset \[STD\]](#)

[IviFgen Frequency \[STD\]](#)

[IviFgen Start Phase \[STD\]](#)

[IviFgen Duty Cycle High \[STD\]](#)

[IviFgen Configure Standard Waveform \[STD\]](#)

When you set this property to `IVIFGEN_VAL_OUTPUT_ARB`, use the following properties and VIs to configure arbitrary waveforms:

[IviFgen Arbitrary Waveform Handle \[ARB\]](#)

[IviFgen Arbitrary Waveform Gain \[ARB\]](#)

[IviFgen Arbitrary Waveform Offset \[ARB\]](#)

[IviFgen Configure Sample Rate \[ARB\]](#)

[IviFgen Create Arbitrary Waveform \[ARB\]](#)

[IviFgen Clear Arbitrary Waveform \[ARB\]](#)

[IviFgen Configure Arbitrary Waveform \[ARB\]](#)

If your instrument allows you to specify the rate at which an entire arbitrary waveform is generated, you can also use the following

properties and VIs to configure arbitrary waveforms:

[IviFgen Arb Frequency \[AF\]](#)

[IviFgen Configure Arb Frequency \[AF\]](#)

When you set this property to IVIFGEN_VAL_OUTPUT_SEQ, use the following properties and VIs to configure sequences:

[IviFgen Arbitrary Sequence Handle \[SEQ\]](#)

[IviFgen Arbitrary Waveform Gain \[ARB\]](#)

[IviFgen Arbitrary Waveform Offset \[ARB\]](#)

[IviFgen Configure Sample Rate \[ARB\]](#)

[IviFgen Create Arbitrary Sequence \[SEQ\]](#)

[IviFgen Clear Arbitrary Sequence \[SEQ\]](#)

[IviFgen Clear Arbitrary Memory \[SEQ\]](#)

[IviFgen Configure Arbitrary Sequence \[SEQ\]](#)

Property Node Path

IviFgen»Basic Operation»Output Mode

Defined Values

[IVIFGEN_VAL_OUTPUT_FUNC](#)

[IVIFGEN_VAL_OUTPUT_ARB](#)

[IVIFGEN_VAL_OUTPUT_SEQ](#)

IviFgen Reference Clock Source

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviFgen Configure Ref Clock Source

Description

Specifies the reference clock source. The function generator derives frequencies and sample rates that it uses to generate waveforms from the source you specify.

Property Node Path

IviFgen»Basic Operation»Reference Clock Source

Defined Values

[IVIFGEN_VAL_REF_CLOCK_INTERNAL](#)

[IVIFGEN_VAL_REF_CLOCK_EXTERNAL](#)

[IVIFGEN_VAL_REF_CLOCK_RTSM_CLOCK](#)

IviFgen Start Phase [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Standard Waveform

Description

This channel-based property specifies the horizontal offset of the standard waveform the function generator produces. You specify this property in degrees of one waveform cycle. For example, a 180 degree phase offset means output generation begins half way through the waveform. A start phase of 360 degrees offsets the output by an entire waveform cycle. It is therefore identical to a start phase of 0 degrees.

You use this property only when the

[IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC. This property does not affect function generator behavior when you set the [IviFgen Waveform \[STD\]](#) property to IVIFGEN_VAL_WFM_DC.

Property Node Path

IviFgen»Standard Function Output»Start Phase [STD]

IviFgen Trigger Source [TRG]

IviFgenTrigger Property [TRG]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Trigger Source

Description

This channel-based property specifies the trigger source. After the function generator receives a trigger, it generates an output signal based on the operation mode.

Property Node Path

IviFgen»Triggering»Trigger Source [TRG]

Defined Values

[IVIFGEN_VAL_EXTERNAL](#)

[IVIFGEN_VAL_INTERNAL_TRIGGER](#)

[IVIFGEN_VAL_TTL0](#)

[IVIFGEN_VAL_TTL1](#)

[IVIFGEN_VAL_TTL2](#)

[IVIFGEN_VAL_TTL3](#)

[IVIFGEN_VAL_TTL4](#)

[IVIFGEN_VAL_TTL5](#)

[IVIFGEN_VAL_TTL6](#)

[IVIFGEN_VAL_TTL7](#)

[IVIFGEN_VAL_ECL0](#)

[IVIFGEN_VAL_ECL1](#)

[IVIFGEN_VAL_PXI_STAR](#)

[IVIFGEN_VAL_RTSL_0](#)

[IVIFGEN_VAL_RTSL_1](#)

[IVIFGEN_VAL_RTSL_2](#)

IVIFGEN_VAL_RTISI_3

IVIFGEN_VAL_RTISI_4

IVIFGEN_VAL_RTISI_5

IVIFGEN_VAL_RTISI_6

IviFgen Waveform [STD]

IviFgenStdFunc Property [STD]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	N/A	IviFgen Configure Standard Waveform

Description

This channel-based property specifies which standard waveform the function generator produces. Refer to the [diagrams of standard waveforms](#).

You use this property only when the [IviFgen Output Mode](#) property is set to IVIFGEN_VAL_OUTPUT_FUNC.

Property Node Path

IviFgen»Standard Function Output»Waveform [STD]

Defined Values

[IVIFGEN_VAL_WFM_SINE](#)

[IVIFGEN_VAL_WFM_SQUARE](#)

[IVIFGEN_VAL_WFM_TRIANGLE](#)

[IVIFGEN_VAL_WFM_RAMP_UP](#)

[IVIFGEN_VAL_WFM_RAMP_DOWN](#)

[IVIFGEN_VAL_WFM_DC](#)

IviFgen Waveform Quantum [ARB]

IviFgenArbWfm Property [ARB]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviFgen Query Arb Waveform Capabilities

Description

The size of each arbitrary waveform must be a multiple of a quantum value. This property returns the quantum value the function generator allows.

For example, when this property returns a value of 8, all waveform sizes must be a multiple of 8. Typically, this value is constant for the function generator.

Property Node Path

IviFgen»Arbitrary Waveform Output»Waveform Quantum [ARB]

IviFgen Defined Values

Expand this book to view an alphabetized list of IviFgen defined values.

IVIFGEN_VAL_AM_EXTERNAL

The function generator uses a waveform from an external source to amplitude modulate the output signal.

Defined Value: 1

IVIFGEN_VAL_AM_INTERNAL

The function generator uses an internally generated waveform to amplitude modulate the output signal.

Defined Value: 0

IVIFGEN_VAL_AM_INTERNAL_RAMP_DOWN

The function generator amplitude modulates the carrier signal with a negative ramp waveform.

Defined Value: 5

IVIFGEN_VAL_AM_INTERNAL_RAMP_UP

The function generator amplitude modulates the carrier signal with a positive ramp waveform.

Defined Value: 4

IVIFGEN_VAL_AM_INTERNAL_SINE

The function generator amplitude modulates the carrier signal with a sinusoid waveform.

Defined Value: 1

IVIFGEN_VAL_AM_INTERNAL_SQUARE

The function generator amplitude modulates the carrier signal with a square waveform.

Defined Value: 2

IVIFGEN_VAL_AM_INTERNAL_TRIANGLE

The function generator amplitude modulates the carrier signal with a triangular waveform.

Defined Value: 3

IVIFGEN_VAL_ECL0

The function generator waits until it receives a trigger on the ECL0 line.

Defined Value: 119

IVIFGEN_VAL_ECL1

The function generator waits until it receives a trigger on the ECL1 line.

Defined Value: 120

IVIFGEN_VAL_EXTERNAL

The function generator does not produce an output signal until it receives a trigger on its external trigger input terminal.

Defined Value: 1

IVIFGEN_VAL_FM_EXTERNAL

The function generator uses a waveform from an external source to frequency modulate the output signal.

Defined Value: 1

IVIFGEN_VAL_FM_INTERNAL

The function generator uses an internally generated waveform to frequency modulate the output signal.

Defined Value: 0

IVIFGEN_VAL_FM_INTERNAL_RAMP_DOWN

The function generator frequency modulates the carrier signal with a negative ramp waveform.

Defined Value: 5

IVIFGEN_VAL_FM_INTERNAL_RAMP_UP

The function generator frequency modulates the carrier signal with a positive ramp waveform.

Defined Value: 4

IVIFGEN_VAL_FM_INTERNAL_SINE

The function generator frequency modulates the carrier signal with a sinusoid waveform.

Defined Value: 1

IVIFGEN_VAL_FM_INTERNAL_SQUARE

The function generator frequency modulates the carrier signal with a square waveform.

Defined Value: 2

IVIFGEN_VAL_FM_INTERNAL_TRIANGLE

The function generator frequency modulates the carrier signal with a triangular waveform.

Defined Value: 3

IVIFGEN_VAL_INTERNAL_TRIGGER

The function generator waits until it receives a trigger from its internal trigger source.

Defined Value: 3

IVIFGEN_VAL_OPERATE_BURST

The function generator produces a discrete number of waveforms when a trigger event occurs.

Defined Value: 1

IVIFGEN_VAL_OPERATE_CONTINUOUS

The function generator produces output continuously.

Defined Value: 0

IVIFGEN_VAL_OUTPUT_ARB

The driver uses the properties and functions of the IviFgenArbWfm Extension Group to configure the function generator's output signal.

Defined Value: 1

IVIFGEN_VAL_OUTPUT_FUNC

The driver uses the properties and functions of the IviFgenStdFunc extension group to configure the function generator's output signal.

Defined Value: 0

IVIFGEN_VAL_OUTPUT_SEQ

The driver uses the properties and functions of the IviFgenArbWfm and IviFgenArbSeq Extension Groups together to configure the function generator's output signal.

Defined Value: 2

IVIFGEN_VAL_PXI_STAR

The function generator waits until it receives a trigger on the PXI Star bus.

Defined Value: 131

IVIFGEN_VAL_REF_CLOCK_EXTERNAL

The function generator receives the reference clock signal from an external source.

Defined Value: 1

IVIFGEN_VAL_REF_CLOCK_INTERNAL

The function generator produces the reference clock signal internally.

Defined Value: 0

IVIFGEN_VAL_REF_CLOCK_RTSM_CLOCK

The function generator receives the reference clock signal from the RTSM clock source.

Defined Value: 101

IVIFGEN_VAL_REF_CLOCK_RTSI_CLOCK

The function generator receives the reference clock signal from the RTSI clock source.

Defined Value: 101

IVIFGEN_VAL_RTSI_0

The function generator waits until it receives a trigger on the RTSI 0 line.

Defined Value: 141

IVIFGEN_VAL_RTSI_1

The function generator waits until it receives a trigger on the RTSI 1 line.

Defined Value: 142

IVIFGEN_VAL_RTSI_2

The function generator waits until it receives a trigger on the RTSI 2 line.

Defined Value: 143

IVIFGEN_VAL_RTSI_3

The function generator waits until it receives a trigger on the RTSI 3 line.

Defined Value: 144

IVIFGEN_VAL_RTSI_4

The function generator waits until it receives a trigger on the RTSI 4 line.

Defined Value: 145

IVIFGEN_VAL_RTSI_5

The function generator waits until it receives a trigger on the RTSI 5 line.

Defined Value: 146

IVIFGEN_VAL_RTSI_6

The function generator waits until it receives a trigger on the RTSI 6 line.

Defined Value: 147

IVIFGEN_VAL_SOFTWARE_TRIG

The function generator does not produce an output signal until the IviFgen_SendSoftwareTrigger function is called.

If you set the IVIFGEN_ATTR_TRIGGER_SOURCE property to this value, the instrument can only be triggered by calling the IviFgen_SendSoftwareTrigger function.

Defined Value: 2

IVIFGEN_VAL_TTL0

The function generator waits until it receives a trigger on the TTL0 line.

Defined Value: 111

IVIFGEN_VAL_TTL1

The function generator waits until it receives a trigger on the TLL1 line.

Defined Value: 112

IVIFGEN_VAL_TTL2

The function generator waits until it receives a trigger on the TTL2 line.

Defined Value: 113

IVIFGEN_VAL_TTL3

The function generator waits until it receives a trigger on the TTL3 line.

Defined Value: 114

IVIFGEN_VAL_TTL4

The function generator waits until it receives a trigger on the TTL4 line.

Defined Value: 115

IVIFGEN_VAL_TTL5

The function generator waits until it receives a trigger on the TTL5 line.

Defined Value: 116

IVIFGEN_VAL_TTL6

The function generator waits until it receives a trigger on the TTL6 line.

Defined Value: 117

IVIFGEN_VAL_TTL7

The function generator waits until it receives a trigger on the TTL7 line.

Defined Value: 118

IVIFGEN_VAL_WFM_DC

Configures the function generator to produce a constant voltage.

Defined Value: 6

IVIFGEN_VAL_WFM_RAMP_DOWN

Configures the function generator to produce a negative ramp waveform.

Defined Value: 5

IVIFGEN_VAL_WFM_RAMP_UP

Configures the function generator to produce a positive ramp waveform.

Defined Value: 4

IVIFGEN_VAL_WFM_SINE

Configures the function generator to produce a sinusoid waveform.

Defined Value: 1

IVIFGEN_VAL_WFM_SQUARE

Configures the function generator to produce a square waveform.

Defined Value: 2

IVIFGEN_VAL_WFM_TRIANGLE

Configures the function generator to produce a triangular waveform.

Defined Value: 3

Properties for Controlling IviFgen Simulation



Note You can set Simulation Attributes in MAX on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** or in the simulation interactive panels. Refer to **National Instruments IVI Driver Help»Configuring Your System** for more information.

Two sets of properties exist for use with the IviFgen Simulation Driver. The following table describes properties that control behavior of the driver. Another table lists [properties that simulate the status](#) of specific driver VIs.

Properties for Controlling IviFgen Simulation

Name	Data Type	Access	Applies to	Description
INTERACTIVE_SIMULATION	Boolean	W	No	Specifies whether to set the driver to interactive mode. In interactive mode, the simulation driver uses pop-up user interface panels for getting information from you. Non-interactive mode is useful when you do not want the interactive panels to interrupt your test program.
SELF_TEST_CODE	Integer	W	No	Specifies the self-test code. When the IviFgen Self-Test VI is called, the self-test result parameter returns this

				value.
SELF_TEST_MSG	String	W	No	Specifies the self-test message. When the IviFgen Self-Test VI is called, the self-test message parameter returns this string.
ERROR_QUERY_CODE	Integer	W	No	Specifies the error-query code. When the IviFgen Error-Query VI is called, the error code parameter returns this value.
ERROR_QUERY_MSG	String	W	No	Specifies the error-query message. When the IviFgen Error-Query VI is called, the error message parameter returns this string.
DRIVER_REV_QUERY	String	W	No	Specifies the instrument

				driver revision message. When the IviFgen Revision Query VI is called, the instrument driver revision parameter returns this string.
INSTR_REV_QUERY	String	W	No	Specifies the firmware revision message. When the IviFgen Revision Query VI is called, the firmware revision parameter returns this string.
SIMULATE_STATUS_CODE	Boolean	W	No	Specifies whether to simulate return codes from the IviFgen driver VIs. Refer to the IviFgen Properties for Status Simulation

			table.
--	--	--	------------------------

Related Topics

[Setting Up Simulated Errors](#)

[Setting Up Simulation for VXI*plug&play* VIs](#)

Properties for Status Simulation in IviFgen

When an IviFgen driver VI is called, the driver queries whether [NISFGEN_ATTR_SIMULATE_STATUS_CODE](#) is enabled. If enabled, the driver gets the appropriate VI's simulation status code. If the value of the simulation status code represents a warning (has a positive value), then it is returned as the VI's return status only if no other error or warning occurred before the VI ends. If the value of the simulation status code represents an error (has a negative value), then it is returned as the VI's return status only if no other error occurred before the VI ends.

The following IviFgen VIs do not support status code simulation:

- IviFgen Get Error
- IviFgen Clear Error
- IviFgen Get Next Interchange Warning
- IviFgen Reset Next Interchange Warning
- IviFgen Clear Next Interchange Warning
- IviFgen Get Next Coercion Record

Each VI supported by the IviFgen class driver has a corresponding property that determines the status code to return when status code simulation is enabled. The following table lists the status code properties and the VI for which they return a value. These properties are all of type ViInt32, non-readable, and non-channel based.



Note In the following table, the literal string NISFGEN precedes all property names.

IviFgen Properties for Status Simulation

Property	VI
INIT_STATUS	IviFgen Initialize , IviFgen Initialize With Options
CLOSE_STATUS	IviFgen Close
RESET_STATUS	IviFgen Reset
SELF_TEST_STATUS	IviFgen Self-Test
ERROR_QUERY_STATUS	IviFgen Error-

	Query
ERROR_MESSAGE_STATUS	IviFgen Error Message
REVISION_QUERY_STATUS	IviFgen Revision Query
RESET_DEFAULT_STATUS	IviFgen Reset With Defaults
DISABLE_STATUS	IviFgen Disable
INVALIDATE_STATUS	IviFgen Invalidate All Attributes
GET_CH_NAME_STATUS	IviFgen Get Channel Name
ABORT_GENERATION_STATUS	IviFgen Abort Generation
CONFIGURE_OPERATION_MODE_STATUS	IviFgen Configure Operation Mode
CONFIGURE_OUTPUT_ENABLED_STATUS	IviFgen Configure Output Enabled
CONFIGURE_OUTPUT_IMPEDANCE_STATUS	IviFgen Configure Output Impedance
CONFIGURE_OUTPUT_MODE_STATUS	IviFgen Configure Output Mode
CONFIGURE_REF_CLOC_SOURCE_STATUS	IviFgen Configure Ref Clock Source
INITIATE_GENERATION_STATUS	IviFgen Initiate Generation
CONFIGURE_STANDARD_WAVEFORM_STATUS	IviFgen Configure Standard Waveform
CLEAR_ARB_WAVEFORM_STATUS	IviFgen Clear Arbitrary Wfm
CONFIGURE_ARB_WAVEFORM_STATUS	IviFgen Configure Arbitrary Wfm

CONFIGURE_SAMPLE_RATE_STATUS	IviFgen Configure Sample Rate
CREATE_ARB_WAVEFORM_STATUS	IviFgen Create Arbitrary Wfm
QUERY_ARB_WFM_CAPABILITIES_STATUS	IviFgen Query Arb Waveform Capabilities
CONFIGURE_ARB_FREQUENCY_STATUS	IviFgen Configure Arb Frequency
CLEAR_ARB_MEMORY_STATUS	IviFgen Clear Arbitrary Memory
CLEAR_ARB_SEQUENCE_STATUS	IviFgen Clear Arbitrary Sequence
CONFIGURE_ARB_SEQUENCE_STATUS	IviFgen Configure Arbitrary Sequence
CREATE_ARB_SEQUENCE_STATUS	IviFgen Create Arbitrary Sequence
QUERY_ARB_SEQ_CAPABILITIES_STATUS	IviFgen Query Arb Sequence Capabilities
CONFIGURE_TRIGGER_SOURCE_STATUS	IviFgen Configure Trigger Source
CONFIGURE_INTERNAL_TRIGGER_RATE_STATUS	IviFgen Configure Internal Trigger Rate
SEND_SOFTWARE_TRIGGER_STATUS	IviFgen Send Software Trigger
CONFIGURE_BURST_COUNT_STATUS	IviFgen Configure Burst Count
CONFIGURE_AM_ENABLED_STATUS	IviFgen Configure AM Enabled
CONFIGURE_AM_INTERNAL_STATUS	IviFgen Configure AM Internal
CONFIGURE_AM_SOURCE_STATUS	IviFgen Configure

	AM Source
CONFIGURE_FM_ENABLED_STATUS	IviFgen Configure FM Enabled
CONFIGURE_FM_INTERNAL_STATUS	IviFgen Configure FM Internal
CONFIGURE_FM_SOURCE_STATUS	IviFgen Configure FM Source



IviPwrMeter Class Driver Reference for LabVIEW

IviPwrMeter VI and Property Reference

[VI Tree](#)

[Properties by Group](#)

[Error and Completion Codes](#)

IviPwrMeter Class Driver Overview

This instrument driver provides programming support for the IviPwrMeter Class. The IviPwrMeter class is designed to support the typical power meter as well as common extended functionality found in more complex instruments. The IviPwrMeter class conceptualizes a power meter as an instrument that can measure the average RF power of an input signal and can be applied to several different instruments. The driver contains all the VIs required in the IviPwrMeter specification defined by the IVI Foundation. This driver requires NI-VISA and the IVI Compliance Package to be installed.

The IviPwrMeter class is divided into a base capability group and several extension capability groups. The base capability group is used to configure a power meter for a typical measurement (this includes setting the units, the auto range mode, the auto averaging mode, and the correction frequency), initiating a measurement, and returning a measured value. The base capability group supports both single and dual channel measurements.

The IviPwrMeter class also contains extension groups that configure the advanced trigger settings, the manual range, the averaging count, and the reference oscillator of the power meter. The class also contains extension groups that perform zero correction and calibration.

IviPwrMeter Extension Capability Groups

The IviPwrMeter class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviPwrMeterChannelAcquisition</u>	Supports power meters that can perform simultaneous measurements on two or more channels and fetch the measurement from each specified channel. Includes properties for enabling a channel for measurement and VIs for acquiring measurements on a specified channel
<u>IviPwrMeterManualRange</u>	Supports power meters that can manually specify the upper and lower limits of the measurement range. Includes VIs for configuring the measurement range.
<u>IviPwrMeterTriggerSource</u>	Supports power meters that can specify a trigger source and a trigger event on which to trigger a measurement. Includes VIs for configuring the trigger source.
<u>IviPwrMeterInternalTrigger</u>	Supports power meters that can trigger internally on the measurement signal. It specifies properties to configure the internal trigger event source, trigger level, and the trigger slope. Includes VIs for configuring these properties.
<u>IviPwrMeterSoftwareTrigger</u>	Supports power meters that can initiate a measurement based on a

software trigger signal. You can send a software trigger to cause the power meter to trigger a measurement.

[IviPwrMeterAveragingCount](#)

Supports power meters that can filter a signal by averaging it a specified number of times in manual averaging mode. Defines an property and VI to specify the averaging count.

[IviPwrMeterZeroCorrection](#)

Supports power meters that can perform a zero correction on an input channel. Defines VIs to perform the zero correction.

[IviPwrMeterDutyCycleCorrection](#)

Supports power meters that perform a duty cycle correction. Includes VIs for enabling and configuring the duty cycle correction.

[IviPwrMeterCalibration](#)

Supports power meters that can perform calibration for a given power sensor. Defines VIs to perform the calibration.

[IviPwrMeterReferenceOscillator](#)

Supports power meters that can enable an internal reference oscillator. Defines properties that configure the reference oscillator and the VIs that set these properties.

Related Topic

[IVI Inherent Class Capabilities](#)

IviPwrMeter Extension Capability Groups

The IviPwrMeter class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviPwrMeterChannelAcquisition</u>	Supports power meters that can perform simultaneous measurements on two or more channels and fetch the measurement from each specified channel. Includes properties for enabling a channel for measurement and VIs for acquiring measurements on a specified channel.
<u>IviPwrMeterManualRange</u>	Supports power meters that can manually specify the upper and lower limits of the measurement range. Includes VIs for configuring the measurement range.
<u>IviPwrMeterTriggerSource</u>	Supports power meters that can specify a trigger source and a trigger event on which to trigger a measurement. Includes VIs for configuring the trigger source.
<u>IviPwrMeterInternalTrigger</u>	Supports power meters that can trigger internally on the measurement signal. It specifies to configure the internal trigger event source, trigger level, and the trigger slope. Includes VIs for configuring these properties.
<u>IviPwrMeterSoftwareTrigger</u>	Supports power meters that can initiate a measurement based on a software trigger signal. You can send

a software trigger to cause the power meter to trigger a measurement.

IviPwrMeterAveragingCount

Supports power meters that can filter a signal by averaging it a specified number of times in manual averaging mode. Defines an property and VI to specify the averaging count.

IviPwrMeterZeroCorrection

Supports power meters that can perform a zero correction on an input channel. Defines VIs to perform the zero correction.

IviPwrMeterDutyCycleCorrection

Supports power meters that perform a duty cycle correction. Includes VIs for enabling and configuring the duty cycle correction.

IviPwrMeterCalibration

Supports power meters that can perform calibration for a given power sensor. Defines VIs to perform the calibration.

IviPwrMeterReferenceOscillator

Supports power meters that can enable an internal reference oscillator. Defines that configure the reference oscillator and the VIs that set these properties.

IviPwrMeterAveragingCount Extension Group

The IviPwrMeterAveragingCount extension capability group supports power meters that can filter a signal by averaging it a specified number of times in manual averaging mode. The IviPwrMeterAveragingCount extension group defines an property and VI to specify the averaging count.

IviPwrMeterAveragingCount VI

[IviPwrMeter_ConfigureAveragingCount](#)

IviPwrMeterAveragingCount Property

[IVIPWRMETER_ATTR_AVERAGING_COUNT](#)

IviPwrMeterBase Extension Group

The IviPwrMeterBase capability group supports power meters that take a single measurement on one channel as well as instruments that can take synchronous measurements on two channels.

The IviPwrMeterBase capability group defines properties and their values to configure the type of measurement and how the measurement is to be performed. These properties include the units, the auto-range mode, the auto-averaging mode, the correction frequency, and the offset. The IviPwrMeterBase capability group also includes VIs for configuring the power meter and for initiating and retrieving measurements.

IviPwrMeterBase Properties

[IVIPWRMETER_ATTR_AVERAGING_AUTO_ENABLED](#)

[IVIPWRMETER_ATTR_CORRECTION_FREQUENCY](#)

[IVIPWRMETER_ATTR_OFFSET](#)

[IVIPWRMETER_ATTR_RANGE_AUTO_ENABLED](#)

[IVIPWRMETER_ATTR_UNITS](#)

Related Topics

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

[IviPwrMeter behavior model](#)

IviPwrMeterCalibration Extension Group

The IviPwrMeterCalibration extension capability group supports power meters that can perform calibration for a given power sensor. The IviPwrMeterCalibration capability group defines functions to perform the calibration.

IviPwrMeterCalibration VIs

[IviPwrMeter_Calibrate](#)

[IviPwrMeter_IsCalibrationComplete](#)

IviPwrMeterChannelAcquisition Extension Group

The IviPwrMeterChannelAcquisition extension capability group supports power meters that can perform simultaneous measurements on two or more channels and fetch the measurement from each specified channel. The IviPwrMeterChannelAcquisition extension capability also includes properties for enabling a channel for measurement and VIs for acquiring measurements on a specified channel.

IviPwrMeterChannelAcquisition VIs

[IviPwrMeter_ConfigureChannelEnabled](#)

[IviPwrMeter_FetchChannel](#)

[IviPwrMeter_ReadChannel](#)

IviPwrMeterChannelAcquisition Property

IVIPWRMETER_ATTR_CHANNEL_ENABLED

IviPwrMeterDutyCycleCorrection Extension Group

The IviPwrMeterDutyCycleCorrection extension capability group supports power meters that perform a duty cycle correction. The IviPwrMeterDutyCycleCorrection extension capability also includes VIs for enabling and configuring the duty cycle correction.

IviPwrMeterDutyCycleCorrection VI

[IviPwrMeter_ConfigureDutyCycleCorrection](#)

IviPwrMeterDutyCycleCorrection Properties

IVIPWRMETER_ATTR_DUTY_CYCLE_CORRECTION

IVIPWRMETER_ATTR_DUTY_CYCLE_CORRECTION_ENABLED

IviPwrMeterInternalTrigger Extension Group

The IviPwrMeterInternalTrigger extension capability group supports power meters that can trigger internally on the measurement signal. The IviPwrMeterInternalTrigger extension capability group also specifies properties to configure the internal trigger event source, trigger level, and the trigger slope. The IviPwrMeterInternalTrigger extension capability group also includes VIs for configuring these properties.

IviPwrMeterInternalTrigger VI

[IviPwrMeter_Configure_Internal_Trigger](#)

IviPwrMeterInternalTrigger Properties

IVIPWRMETER_ATTR_INTERNAL_TRIGGER_EVENT_SOURCE

IVIPWRMETER_ATTR_INTERNAL_TRIGGER_LEVEL

IVIPWRMETER_ATTR_INTERNAL_TRIGGER_SLOPE

IviPwrMeterManualRange Extension Group

The IviPwrMeterManualRange extension capability group supports power meters that can manually specify the upper and lower limits of the measurement range. The IviPwrMeterManualRange extension capability also includes VIs for configuring the measurement range.

IviPwrMeterManualRange VI

[IviPwrMeter_ConfigureRange](#)

IviPwrMeterManualRange Properties

IVIPWRMETER_ATTR_RANGE_LOWER

IVIPWRMETER_ATTR_RANGE_UPPER

IviPwrMeterReferenceOscillator Extension Group

The IviPwrMeterReferenceOscillator extension capability group supports power meters that can enable an internal reference oscillator. The IviPwrMeterReferenceOscillator capability group defines properties that configure the reference oscillator. It also defines VIS to set these properties.

IviPwrMeterReferenceOscillator VIs

[IviPwrMeter_ConfigureRefOscillator](#)

[IviPwrMeter_ConfigureRefOscillatorEnabled](#)

IviPwrMeterReferenceOscillator Properties

IVIPWRMETER_ATTR_REF_OSCILLATOR_ENABLED

IVIPWRMETER_ATTR_REF_OSCILLATOR_FREQUENCY

IVIPWRMETER_ATTR_REF_OSCILLATOR_LEVEL

IviPwrMeterSoftwareTrigger Extension Group

The IviPwrMeterSoftwareTrigger extension group supports power meters that can initiate a measurement based on a software trigger signal. You can send a software trigger to cause the power meter to trigger a measurement.

IviPwrMeterSoftwareTrigger VI

[IviPwrMeter_SendSoftwareTrigger](#)

IviPwrMeterTriggerSource Extension Group

The IviPwrMeterTriggerSource extension capability group supports power meters that can specify a trigger source and a trigger event on which to trigger a measurement. The IviPwrMeterTriggerSource extension capability also includes VIs for configuring the trigger source.

IviPwrMeterTriggerSource VI

[IviPwrMeter_Configure_Trigger_Source](#)

IviPwrMeterTriggerSource Property

[IVIPWRMETER_ATTR_TRIGGER_SOURCE](#)

IviPwrMeterZeroCorrection Extension Group

The IviPwrMeterZeroCorrection extension capability group supports power meters that can perform a zero correction on an input channel. The IviPwrMeterZeroCorrection capability group defines VIs to perform the zero correction.

IviPwrMeterZeroCorrection VIs

[IviPwrMeter_Is_Zero_Correction_Complete](#)

[IviPwrMeter_Zero](#)

[IviPwrMeter_ZeroAllChannels](#)

IviPwrMeter Behavior Models

[IviPwrMeterAveragingCount](#)

[IviPwrMeterBase](#)

[IviPwrMeterCalibration](#)

[IviPwrMeterChannelAcquisition](#)

[IviPwrMeterDutyCycleCorrection](#)

[IviPwrMeterInternalTrigger](#)

[IviPwrMeterManualRange](#)

[IviPwrMeterReferenceOscillator](#)

[IviPwrMeterSoftwareTrigger](#)

[IviPwrMeterTriggerSource](#)

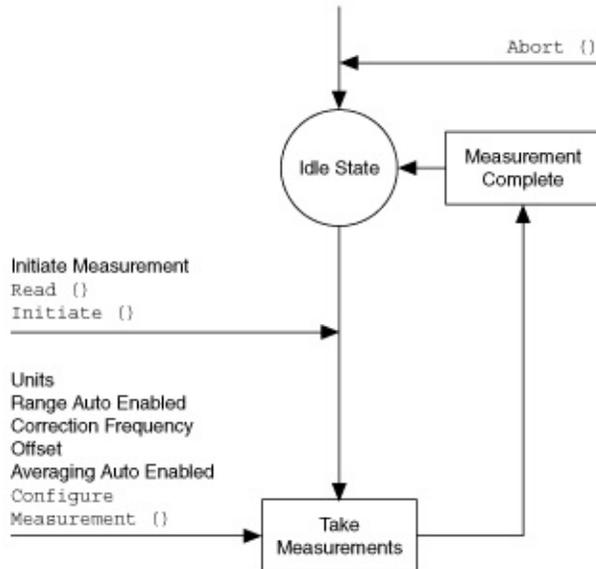
[IviPwrMeterZeroCorrection](#)

IviPwrMeterAveragingCount Behavior Model

The IviPwrMeterAveragingCount extension group follows the same behavior model as the [IviPwrMeterBase](#) capability group.

IviPwrMeterBase Behavior Model

The following state diagram shows relationships between the [IviPwrMeterBase](#) capability group and power meter behavior.



The main state in the IviPwrMeter Class is the Idle state. The power meter enters the Idle state as the result of being "powered-on", successfully completing a measurement, or by being aborted from a previous measurement by you with the [Abort VI](#). Typically, you configure the power meter while it is in the Idle state. IviPwrMeter attributes can be configured individually by setting properties or with the high-level configuration VIs defined in the [IviPwrMeterBase](#) capability group.

The [Read](#) and [Initiate](#) VIs cause the power meter to leave the Idle state and take a measurement on all enabled channels. The [Read](#) VI does not return until the measurement process is complete and the power meter has returned to the Idle state. The [Initiate](#) VI returns as soon as the power meter leaves the Idle state. The [Fetch](#) VI is used to retrieve measurements that were initiated by the [Initiate](#) VI.

As soon as the power meter leaves the Idle state, it immediately takes a measurement on all enabled channels. You enable channels by calling the [Configure Measurement VI](#) before initiating the measurement. The power meter takes a measurement on the channels specified in the [Configure Measurement VI](#) and performs the specified math operation on the results.

After all measurements have been taken, the power meter (if it is capable

of doing so) generates the Measurement Complete signal and returns to the Idle state. The IviPwrMeterBase capability group does not require that a power meter be able to generate a Measurement Complete signal. The IviPwrMeterBase capability group does not define how a Measurement Complete signal is configured. The Measurement Complete signal is presented in the IviPwrMeter behavior model diagram to define when the signal is generated as most power meters generate this signal but may not be able to configure it.

IviPwrMeterCalibration Behavior Model

The IviPwrMeterCalibration extension group follows the same behavior model as the [IviPwrMeterBase](#) capability group.

IviPwrMeterChannelAcquisition Behavior Model

The IviPwrMeterChannelAcquisition behavior model leverages the behavior model of the [IviPwrMeterBase](#) capability group.

IviPwrMeterDutyCycleCorrection Behavior Model

The IviPwrMeterDutyCycleCorrection extension group follows the same behavior model as the [IviPwrMeterBase](#) capability group.

IviPwrMeterInternalTrigger Behavior Model

The IviPwrMeterInternalTrigger behavior model leverages the behavior model of the [IviPwrMeterBase](#) capability group and the [IviPwrMeterTriggerSource](#) extension group. Furthermore, it defines an additional trigger event for the trigger source.

The power meter leaves the Wait-For-Trigger state when it receives an internal trigger event. When the [Trigger Source](#) property is set to the [Internal](#) defined value, and the measurement signal on the channel specified by the [Internal Trigger Event Source](#) property crosses the level specified by the [Internal Trigger Level](#) property in the direction specified by the [Internal Trigger Slope](#) property, the power meter leaves the Wait-For-Trigger state and takes a measurement on all enabled channels.

IviPwrMeterManualRange Behavior Model

The IviPwrMeterManualRange behavior model leverages the behavior model of the [IviPwrMeterBase](#) capability group.

IviPwrMeterReferenceOscillator Behavior Model

The IviPwrMeterReferenceOscillator extension group follows the same behavior model as the [IviPwrMeterBase](#) capability group.

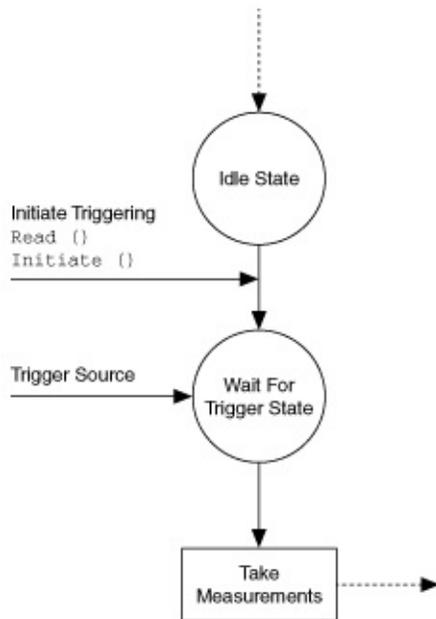
IviPwrMeterSoftwareTrigger Behavior Model

The IviPwrMeterSoftwareTrigger behavior model leverages the behavior model of the [IviPwrMeterBase](#) capability group and the [IviPwrMeterTriggerSource](#) extension group. Furthermore, it defines an additional trigger event for the trigger source.

The power meter leaves the Wait-For-Trigger state when it receives a trigger event specified by the [Trigger Source](#) property. When the trigger source is set to Software, the [Send Software Trigger VI](#) is used to generate the trigger event. Calling this VI causes the power meter to leave the Wait-For-Trigger state and take a measurement on all enabled channels.

IviPwrMeterTriggerSource Behavior Model

The IviPwrMeterTriggerSource behavior model leverages the behavior model of the [IviPwrMeterBase](#) capability group. Furthermore, it defines an additional Wait-For-Trigger state after the Idle state. The following state diagram shows the relationship between IviPwrMeterTriggerSource capabilities and the IviPwrMeterBase capability behavior model.



The [Initiate](#) and [Read](#) VIs cause the power meter to leave the Idle state and transition to the Wait-For-Trigger state. The [Read VI](#) does not return until the measurement process is complete and the power meter returns to the Idle state. The [Initiate VI](#) returns as soon as the power meter leaves the Idle state.

In the Wait-For-Trigger state, the power meter waits for a trigger event. The type of trigger event is specified by the [Trigger Source](#) property. When the specified trigger event occurs, the power meter leaves the Wait-For-Trigger state and takes a measurement on all enabled channels.

IviPwrMeterZeroCorrection Behavior Model

The IviPwrMeterZeroCorrection extension group follows the same behavior model as the [IviPwrMeterBase](#) capability group.

IviPwrMeter VI Tree

Class/Panel Name	VI Name
Configuration	
Configure Units	<u>IviPwrMeter_Configure Units.vi</u>
Configure Measurement	<u>IviPwrMeter_Configure Measurement.vi</u>
Configure Range Auto Enabled	<u>IviPwrMeter_Configure Range Auto Enabled.vi</u>
Configure Averaging Auto Enabled	<u>IviPwrMeter_Configure Averaging Auto Enabled.vi</u>
Configure Correction Frequency	<u>IviPwrMeter_Configure Correction Frequency.vi</u>
Configure Offset	<u>IviPwrMeter_Configure Offset.vi</u>
Configure Range [RNG]	<u>IviPwrMeter_Configure Range [RNG].vi</u>
Configure Averaging Count [AVG]	<u>IviPwrMeter_Configure Averaging Count [AVG].vi</u>
Configure Duty Cycle Correction [DC]	<u>IviPwrMeter_Configure Duty Cycle Correction [DC].vi</u>
Get Channel Name	<u>IviPwrMeter_Get Channel Name.vi</u>
Configure Channel Enabled [CH]	<u>IviPwrMeter_Configure Channel Enabled [CH].vi</u>
Trigger	
Configure Trigger Source [TRG]	<u>IviPwrMeter_Configure Trigger Source [TRG].vi</u>
Configure Internal Trigger [IT]	<u>IviPwrMeter_Configure Internal Trigger [IT].vi</u>

Configure Internal Trigger Level [IT]

[IviPwrMeter_Configure
Internal Trigger Level
\[IT\].vi](#)

Reference Oscillator

Configure Reference Oscillator Enabled [RO]

[IviPwrMeter_Configure
Reference Oscillator
Enabled \[RO\].vi](#)

Configure Reference Oscillator [RO]

[IviPwrMeter_Configure
Reference Oscillator
\[RO\].vi](#)

Zeroing

Zero [ZC]

[IviPwrMeter_Zero \[ZC\].vi](#)

Zero All Channels [ZC]

[IviPwrMeter_Zero All
Channels \[ZC\].vi](#)

Is Zero Correction Complete [ZC]

[IviPwrMeter_Is Zero
Correction Complete
\[ZC\].vi](#)

Calibration

Calibrate [CAL]

[IviPwrMeter_Calibrate
\[CAL\].vi](#)

Is Calibration Complete [CAL]

[IviPwrMeter_Is
Calibration Complete
\[CAL\].vi](#)

Measurement

Read

[IviPwrMeter_Read.vi](#)

Read Channel [CH]

[IviPwrMeter_Read
Channel \[CH\].vi](#)

Low-Level Measurement

Initiate

[IviPwrMeter_Initiate.vi](#)

Is Measurement Complete

[IviPwrMeter_Is
Measurement
Complete.vi](#)

Fetch

[IviPwrMeter_Fetch.vi](#)

Fetch Channel [CH]

[IviPwrMeter_Fetch
Channel \[CH\].vi](#)

Query Result Range Type

[IviPwrMeter_Query
Result Range Type.vi](#)

Abort

[IviPwrMeter_Abort.vi](#)

Send Software Trigger [SWT]

[IviPwrMeter_Send
Software Trigger \[SWT\].vi](#)

IviPwrMeter Properties

Group/Attribute Name	Property Label
Basic Operation	
Averaging Auto Enabled	<u>IviPwrMeter»Basic Operation»Averaging Auto Enabled</u>
Correction Frequency	<u>IviPwrMeter»Basic Operation»Correction Frequency</u>
Offset	<u>IviPwrMeter»Basic Operation»Offset</u>
Range Auto Enabled	<u>IviPwrMeter»Basic Operation»Range Auto Enabled</u>
Units	<u>IviPwrMeter»Basic Operation»Units</u>
Channel Acquisition	
Channel Enabled [CH]	<u>IviPwrMeter»Basic Operation»Channel Acquisition»Channel Enabled [CH]</u>
Manual Range	
Range Lower [RNG]	<u>IviPwrMeter»Manual Range»Range Lower [RNG]</u>
Range Upper [RNG]	<u>IviPwrMeter»Manual Range»Range Upper [RNG]</u>
Trigger	
Trigger Source [TRG]	<u>IviPwrMeter»Trigger»Trigger Source [TRG]</u>
Internal Trigger	
Internal Trigger Event Source [IT]	<u>IviPwrMeter»Trigger»Internal Trigger»Internal Trigger Event Source [IT]</u>
Internal Trigger Level [IT]	<u>IviPwrMeter»Trigger»Internal Trigger»Internal Trigger Level [IT]</u>

Internal Trigger Slope [IT]

[IviPwrMeter»Trigger»Internal Trigger»Internal Trigger Slope \[IT\]](#)

Duty Cycle

Duty Cycle Correction [DC]

[IviPwrMeter»Duty Cycle»Duty Cycle Correction \[DC\]](#)

Duty Cycle Correction Enabled [DC]

[IviPwrMeter»Duty Cycle»Duty Cycle Correction Enabled \[DC\]](#)

Averaging Count

Average Count [AVG]

[IviPwrMeter»Averaging Count»Average Count \[AVG\]](#)

Reference Oscillator

Ref Oscillator Enabled [RO]

[IviPwrMeter»Reference Oscillator»Ref Oscillator Enabled \[RO\]](#)

Ref Oscillator Frequency [RO]

[IviPwrMeter»Reference Oscillator»Ref Oscillator Frequency \[RO\]](#)

Ref Oscillator Level [RO]

[IviPwrMeter»Reference Oscillator»Ref Oscillator Level \[RO\]](#)

IviPwrMeter Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviPwrMeter Errors	0xBFFFA2001 to 0xBFFFA3FFF
IviPwrMeter Warnings	0x3FFFA2001 to 0x3FFFA3FFF
IVI Specific Driver Errors	0xBFFFA4000 to 0xBFFFA5FFF
IVI Specific Driver Warnings	0x3FFFA4000 to 0x3FFFA5FFF
IVI Errors	0xBFFFA0000 to 0xBFFFA1FFF
IVI Warnings	0x3FFFA0000 to 0x3FFFA1FFF
Common Instrument Driver Errors	0xBFFFC0000 to 0xBFFFCFFFF
Common Instrument Driver Warnings	0x3FFFC0000 to 0x3FFFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0xFFFFFFF

The IviPwrMeter class driver defines the error codes shown in the following table in addition to the IVI defined error codes.

IviPwrMeter Error Codes

Error	Value	Message
IVIPWRMETER_ERROR_CHANNEL_NOT_ENABLED	0xBFFFA2001	Character name enabled measurement
IVIPWRMETER_ERROR_MAX_TIME_EXCEEDED	0xBFFFA2020	Maximum time exceeded before operation could complete
IVIPWRMETER_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFFA1001	Maximum time exceeded before operation could complete

IviPwrMeter Warning Code

Warning	Value	Message
IVIPWRMETER_WARN_UNDER_RANGE	0x3FFA2001	Under Range warning
IVIPWRMETER_WARN_OVER_RANGE	0x3FFA2002	Over Range warning

Related Topic

[IVI Status Codes](#)

IviPwrMeter VIs

Expand this book to view an alphabetized list of IviPwrMeter VIs.

IviPwrMeter Abort

IviPwrMeterBase Capability Group

This VI aborts all previously initiated measurements and returns the power meter to the Idle state. You initiate measurement with the .



Notes

(1) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

(2) If the instrument cannot abort an initiated acquisition, this VI returns the `IVI_ERROR_FUNCTION_NOT_SUPPORTED` error.

 **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

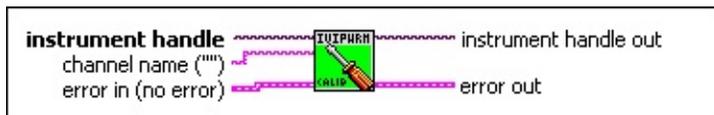
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Calibrate [CAL]

IviPwrMeterCalibration Capability Group

This VI performs calibration on the specified sensor. This VI returns only after the sensor has been calibrated. You may use the to determine when the calibration is complete.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name of the channel to calibrate.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- EEH** **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- I/O** **instrument handle out** The instrument handle that you obtain

from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



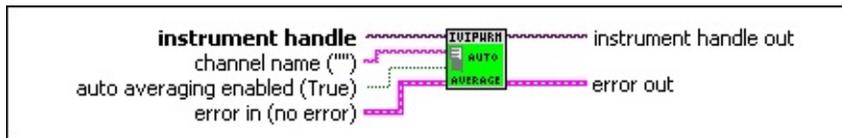
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Averaging Auto Enabled

IviPwrMeterBase Capability Group

This VI enables or disables the auto-averaging mode for a given channel.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name of the channel to be configured.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- TF** **auto averaging enabled** The auto-averaging mode. Pass TRUE to turn auto-averaging on. Pass FALSE to turn auto-averaging off. The driver sets the property to this value.

Valid Values:

TRUE - "On"

FALSE - "Off"

Default Value:

TRUE - "On"

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

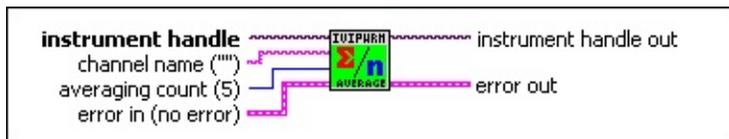
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Averaging Count [AVG]

IviPwrMeterAveragingCount Capability Group

This VI sets the average count that the instrument uses in manual averaging mode. The averaging count specifies the number of samples that the instrument takes before the measurement is complete.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **channel name** The name of the channel to configure.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

I32 **averaging count** Specifies the averaging count. The driver sets the property to this value.

Default Value: 5

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

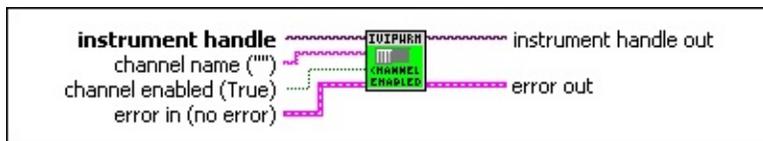
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Channel Enabled [CH]

IviPwrMeterChannelAcquisition Capability Group

This VI enables or disables a specified channel for measurement.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **channel name** Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

TF **channel enabled** Pass TRUE to enable the channel. Pass FALSE to disable the channel. The driver sets the property to this value.

Valid Values:

TRUE - "On"

FALSE - "Off"

Default Value:

TRUE - "On"

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

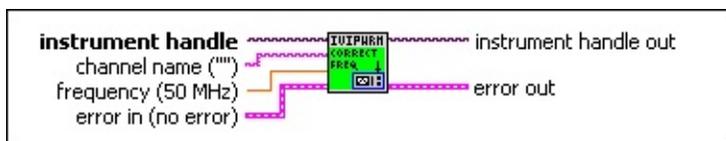
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Correction Frequency

IviPwrMeterBase Capability Group

This VI specifies the frequency of the input signal in Hertz. The instrument uses this value to determine the appropriate correction factor for the sensor.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name of the channel to be configured.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- DBL** **frequency** Specifies the expected frequency of the input signal in Hertz. The instrument uses this parameter to determine the appropriate correction factor for the sensor. The driver sets the property to this value.

Units: Hz

Default Value: 50 MHz

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

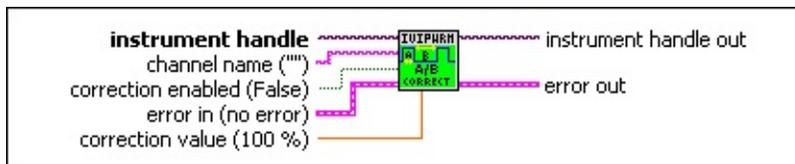
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Duty Cycle Correction [DC]

IviPwrMeterDutyCycleCorrection Capability Group

This VI enables or disables the duty cycle correction and sets the duty cycle correction for pulse power measurements.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **channel name** The name of the channel for which to specify the duty cycle correction.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

TF **correction enabled** Enables or disables the duty cycle correction. The driver sets the `_ENABLED` property to this value.

Valid Values:
TRUE - "On"
FALSE - "Off"

Default Value:
FALSE - "Off"

 **correction value** Specifies the duty cycle correction the power meter uses to calculate the pulse power of a pulse-modulated signal. The driver sets the property to this value.

The power meter measures the average power of the pulsed input signal and then divides the result by the duty cycle correction value to obtain a pulse power reading.

The value of this property is specified as a percentage.
For a pulse with a duty cycle of 10%, this property should be given the value 0.1.

Units: percentage (%).

Default Value: 100 %

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

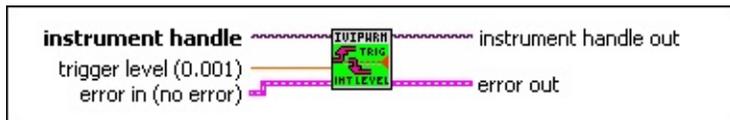
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Internal Trigger Level [IT]

IviPwrMeterInternalTrigger Capability Group

This VI configures the internal trigger level of the power meter.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

DBL **trigger level** The signal trigger level. The driver sets the property to this value.

The value of this property is specified in the same unit as the value of the property which you can set by calling the .

Default Value: 0.001

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

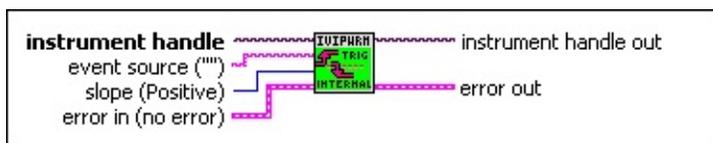
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Internal Trigger [IT]

IviPwrMeterInternalTrigger Capability Group

This VI configures the internal trigger event source and the internal trigger slope of the power meter.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **event source** The name of the channel to use as the internal trigger event source. The driver sets the property to this value.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

I32 **slope** The internal trigger slope. The driver sets the property to this value.

Valid Values:

IVIPWRMETER_VAL_POSITIVE - Sets the trigger event to occur on the rising edge of the trigger pulse.

IVIPWRMETER_VAL_NEGATIVE (Difference) - Sets the trigger event to occur on the falling edge of the trigger pulse.

Default Value:

IVIPWRMETER_VAL_POSITIVE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Measurement

IviPwrMeterBase Capability Group

This VI configures the instrument to take single or dual channel measurements.

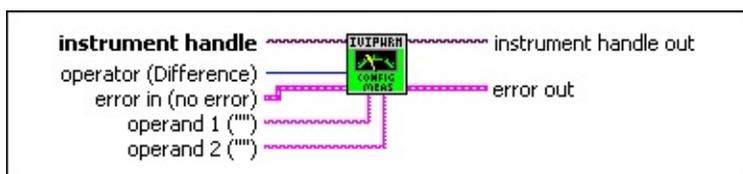
For single channel measurements, this VI enables the channel specified by Operand1 and disables all other channels. The result returned by the Fetch or Read VIs is the measurement taken at the channel specified by Operand1. Although, the driver measures the power in Watts, the result is converted to the same unit as the value of the property.

For dual channel measurements, this VI enables the channels specified by Operand1 and Operand2 and disables all other channels. The result returned by the Fetch or Read VIs is the result of the specified math operation applied to the measurements on the channels specified by Operand1 and Operand2.

Although, the math operation is performed on the measured values in Watts, the result is converted to the appropriate units depending on the value of the Units property and the value of the Operator.

For Difference and Sum operations, the resulting units is the same as the Units property.

For Quotient operations, the resulting units are in dB, except when Units are set to Watts. When set to Watts, the resulting measurement is without units.



I/O **instrument handle** The instrument handle that you obtain from the

[IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **operator** The math VI applied to the operands.

To configure the power meter to take single channel measurements, pass None as the Operator value. To configure the power meter to take simultaneous dual channel measurements, pass one of the other defined math operators as the Operator value.

Valid Values:

IVIPWRMETER_VAL_NONE (No Operator) - Take a single channel measurement. Return the measured value for Operand1. Operand2 is ignored.

IVIPWRMETER_VAL_DIFFERENCE (Difference) - Subtract the power measured on Operand2 from the power measured on Operand1.

IVIPWRMETER_VAL_SUM (Sum) - Add the power measured on Operand2 to the power measured on Operand1.

IVIPWRMETER_VAL_QUOTIENT (Quotient) - Divide the power measured on Operand1 by the power measured on Operand2.

Default Value:

IVIPWRMETER_VAL_DIFFERENCE (Difference)

 **operand 1** The name of the channel from which the value for the first operand of the math operator is measured.

Default Value: ""

 **operand 2** The name of the channel from which the value for the second operand of the math operator is measured.

Default Value: ""

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if

any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



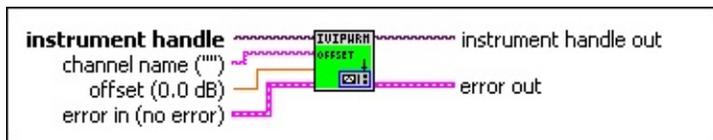
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Offset

IviPwrMeterBase Capability Group

This VI specifies the offset to be added to the measured value in units of dB.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name of the channel for which to set the Offset.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- DBL** **offset** Specifies an offset to be added to the measured value. The units of this property are dB. This property can be used to compensate for system losses or gains between the unit under test and the sensor of the power meter. The driver sets the property to this value.

For example, a cable loss of 2 dB could be compensated for by

setting this property to 2. Similarly, a gain stage of 10 dB could be accounted for by setting the value of this property to -10. In both cases, the reading from the power meter will indicate the power at the unit under test rather than power at the power meter's sensor.

Unit: dB

Default Value: 0.0 dB



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



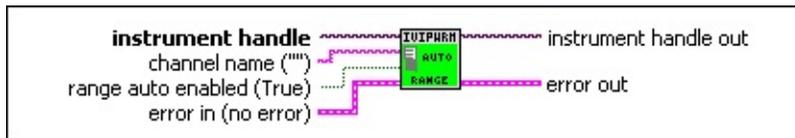
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Range Auto Enabled

IviPwrMeterBase Capability Group

This VI enables or disables the auto range mode for a given channel.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name of the channel to configure.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- TF** **range auto enabled** Specifies the auto range mode. Pass TRUE to turn auto ranging on. Pass FALSE to turn auto ranging off. The driver sets the property to this value.

Valid Values:

TRUE - "On"

FALSE - "Off"

Default Value:

TRUE - "On"

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

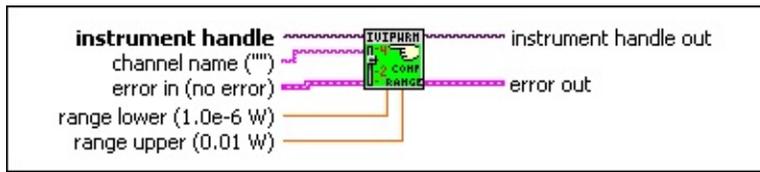
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Range [RNG]

IviPwrMeterManualRange Capability Group

This VI configures lower and upper range values for a given channel.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **channel name** The name of the channel to configure.

Pass the virtual channel name that you assign to the instrument in the Configuration Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

DBL **range lower** Specify the lower limit of the expected value of the measurement. The driver sets the `IviPwrMeter>>Manual Range>>Range Lower [R]` property to this value.

Unit: Watt

Default Value: 1.0e-6 (W)



range upper Specify the upper limit of the expected value of the measurement. The driver sets the `IviPwrMeter>>Manual Range>>Range Upper [R]` property to this value.

Unit: Watt

Default Value: 0.01 (W)



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



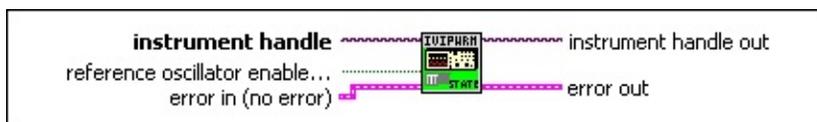
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Reference Oscillator Enabled [RO]

IviPwrMeterReferenceOscillator Capability Group

This VI enables or disables the reference oscillator.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

TF **reference oscillator enabled** Pass TRUE to enable the reference oscillator. Pass FALSE to disable the reference oscillator. The driver sets the property to this value.

Valid Values:

TRUE - "On"

FALSE - "Off"

Default Value:

TRUE - "On"

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

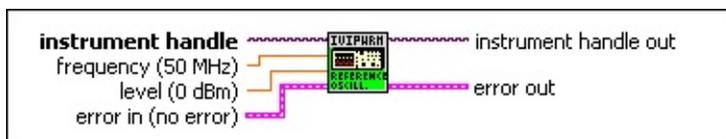
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Reference Oscillator [RO]

IviPwrMeterReferenceOscillator Capability Group

This VI sets the frequency and power level of the signal generated by the reference oscillator.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

DBL **frequency** The frequency of the reference oscillator. The driver sets the property to this value.

Units: Hertz.

Default Value: 50 MHz

DBL **level** The power level of the reference oscillator. The driver sets the property to this value.

Units: dBm.

Default Value: 0 dBm

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O

instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



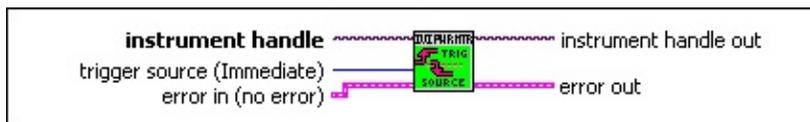
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Trigger Source [TRG]

IviPwrMeterTriggerSource Capability Group

This VI configures the trigger source of the power meter.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

I32 **trigger source** Specifies the trigger source. After you call either the or the , the Power Meter waits for the trigger you specify. The driver sets the property to this value.

Defined Values:

IVIPWRMETER_VAL_IMMEDIATE (Immediate) - The power meter exits the Wait-For-Trigger state immediately after entering. It does not wait for a trigger of any kind.

IVIPWRMETER_VAL_EXTERNAL (External) - The power meter exits the Wait-For-Trigger state when an internal trigger event occurs on the measurement signal.

IVIPWRMETER_VAL_INTERNAL (Internal) - The power meter exits the Wait-For-Trigger state when a trigger occurs on the external trigger input.

IVIPWRMETER_VAL_SOFTWARE_TRIG (Software) - The power meter exits the Wait-For-Trigger state when it receives software trigger.

IVIPWRMETER_VAL_TTL0 (TTL0) - The power meter exits the Wait-For-Trigger state when it receives a

trigger on TTL0.

IVIPWRMETER_VAL_TTL1 (TTL1) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL1.

IVIPWRMETER_VAL_TTL2 (TTL2) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL2.

IVIPWRMETER_VAL_TTL3 (TTL3) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL3.

IVIPWRMETER_VAL_TTL4 (TTL4) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL4.

IVIPWRMETER_VAL_TTL5 (TTL5) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL5.

IVIPWRMETER_VAL_TTL6 (TTL6) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL6.

IVIPWRMETER_VAL_TTL7 (TTL7) - The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL7.

IVIPWRMETER_VAL_ECL0 (ECL0) - The power meter exits the Wait-For-Trigger state when it receives a trigger on ECL0.

IVIPWRMETER_VAL_ECL1 (ECL1) - The power meter exits the Wait-For-Trigger state when it receives a trigger on ECL1.

IVIPWRMETER_VAL_PXI_STAR (PXI Star) - The power meter

exits the Wait-For-Trigger state when it receives a trigger on PXI Star trigger bus.

IVIPWRMETER_VAL_RTISI_0 (RTSI0) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI0 line.

IVIPWRMETER_VAL_RTISI_1 (RTSI1) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI1 line.

IVIPWRMETER_VAL_RTISI_2 (RTSI2) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI2 line.

IVIPWRMETER_VAL_RTISI_3 (RTSI3) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI3 line.

IVIPWRMETER_VAL_RTISI_4 (RTSI4) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI4 line.

IVIPWRMETER_VAL_RTISI_5 (RTSI5) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI5 line.

IVIPWRMETER_VAL_RTISI_6 (RTSI6) - The power meter exits the Wait-For-Trigger state when it receives a trigger on RTSI6 line.

Default Value:

IVIPWRMETER_VAL_IMMEDIATE (Immediate)



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



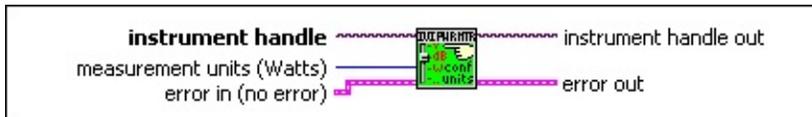
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Configure Units

IviPwrMeterBase Capability Group

This VI configures the unit to which the RF power is converted after measurement.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- I32** **measurement units** Specifies the units at the input terminals of the power meter. The driver sets the property to this value.

Defined Values:

- IVIPWRMETER_VAL_DBM (dBm)
- IVIPWRMETER_VAL_DBMV (dBmV)
- IVIPWRMETER_VAL_DBUV (dBuV)
- IVIPWRMETER_VAL_WATTS (Watts)

Default Value:

- IVIPWRMETER_VAL_WATTS (Watts)

- Err** **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- I/O** **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- Err** **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

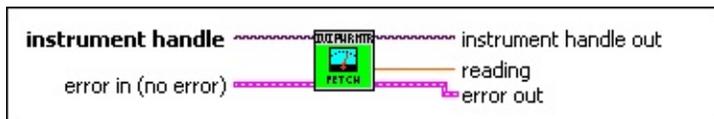
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Fetch

IviPwrMeterBase Capability Group

This VI returns the result from a previously initiated single or dual channel measurement. Call the [IviPwrMeter Initialize](#) to initiate a measurement before calling this VI.

After this VI executes, the value of Reading depends on the math operation specified in the [IviPwrMeter Initialize](#).



Notes

(1) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviPwrMeter Status](#) at the conclusion of the sequence.

(2) If an out-of-range condition occurs on one or more enabled channels, the result is a value indicating that an out of range condition occurred. In such a case, Reading contains an IEEE defined -Inf (Negative Infinity) or +Inf (Positive Infinity) value and the VI returns the Under Range (0x3FFA2001) or Over Range (0x3FFA2002) warning. Test if the measurement value is out of range with the [IviPwrMeter Is Out of Range](#).

I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



reading Returns the measured value.

For single channel measurements, Reading contains an actual reading on the channel specified by the . The unit of the result is the same as the value of the Units property.

For dual channel measurements, Reading contains the result of the math operation applied to the channels specified in the . The unit of the result depends on the value of the property and the specified operator.

For Difference and Sum operations, the resulting units is the same as the Units property.

For Quotient operations, the resulting units are in dB, except when Units are set to Watts. When set to Watts, the resulting measurement is without units.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

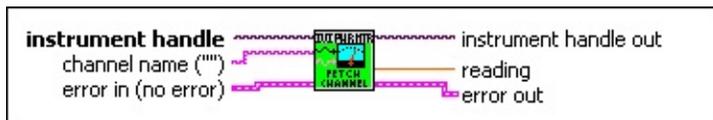
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Fetch Channel [CH]

IviPwrMeterChannelAcquisition Capability Group

This VI returns the result from a previously initiated measurement on a specified channel. Call the [IviPwrMeter Channel Acquisition](#) to initiate a measurement before calling this VI.

After this VI executes, the Reading parameter contains an actual reading on the channel specified by the Channel parameter. If the specified channel is not enabled for measurement, this VI returns the Channel Not Enabled (0xBFFA2001) error. The result is in the same unit as the value of the Units property.



Notes

(1) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviPwrMeter Status](#) at the conclusion of the sequence.

(2) If an out of range condition occurs, the result is a value indicating that an out-of-range condition occurred. In such a case, the Reading parameter contains an IEEE defined -Inf (Negative Infinity) or +Inf (Positive Infinity) value and the VI returns the Under Range (0x3FFA2001) or Over Range (0x3FFA2002) warning. Test if the measurement value is out of range with the [IviPwrMeter Is Out of Range](#).

I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



channel name The name of the channel from which to fetch the measurement.

Pass the virtual channel name that you assign to the instrument in the Action Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



reading The data read from the power meter.



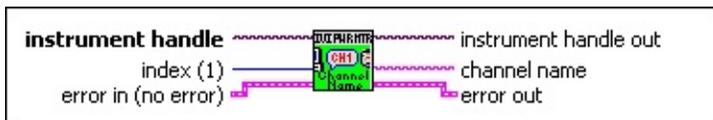
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Get Channel Name

IviPwrMeterBase Capability Group

This VI returns the physical channel identifier that corresponds to the one-based index specified by the ChannelIndex parameter.



Notes

(1) If you pass in a value for the ChannelIndex parameter that is less than one or greater than the value of the Channel Count property, the VI returns an empty string in the ChannelName parameter and returns an error.

(2) By passing 0 for the buffer size, you can ascertain the buffer size required to get the entire channel name string and then call the VI again with a sufficiently large buffer.

I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

I32 **index** A 1-based index into the channel table.

Valid Range: 1 to number of channels.

Default Value: 1

I32 **buffer size** Pass the number of bytes in the ViChar array you specify for the Channel Name parameter.

If the channel name, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the VI copies BufferSize - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to

get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the VI places "123" into the buffer and returns 7.

If you pass a negative number, the VI copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI_NULL for the Channel Name buffer parameter.

Default Value: None

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **channel name** Returns the highest-level channel name that corresponds to the specific driver channel string that is in the channel table at an index you specify..

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the channel name description, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the VI copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the VI places "123" into the buffer and returns 7.

If you pass 0 for the Buffer Size, you can pass VI_NULL for this parameter.

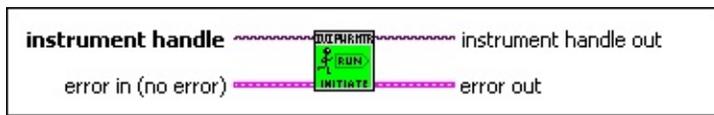
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Initiate

IviPwrMeterBase Capability Group

This VI initiates a measurement on all enabled channels. When this VI executes, the power meter leaves the Idle state and takes a measurement on all enabled channels. Use the or to obtain the result of the measurements.



Notes

(1) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

(2) This VI performs interchangeability checking when the IviPwrMeter>>Inherent IVI Settings>>User Options>>Interchange Check property is set to TRUE. You may use the NI Spy utility to view interchangeability warnings. Alternatively, you may use the to retrieve interchangeability warnings. For information on interchangeability checking, refer to the online help manual.

 **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Is Calibration Complete [CAL]

IviPwrMeterCalibration Capability Group

This VI queries the instrument to determine the status of all calibration operations initiated by the . This VI returns the `IVIPWRMETER_VAL_CALIBRATION_COMPLETE` (1) value in the Status parameter only when calibration is complete on all channels.

If some calibration operations are still in progress on one or more channels, the driver returns the `IVIPWRMETER_VAL_CALIBRATION_IN_PROGRESS` (0) value. If the driver cannot query the instrument to determine its state, the driver returns the `IVIPWRMETER_VAL_CALIBRATION_STATUS_UNKNOWN` (-1) value.

Note:

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **calibration status** Returns the status of the calibration.

This driver defines the following calibration status:

IVIPWRMETER_VAL_CALIBRATION_COMPLETE (1)

- The power meter has completed the calibration on all enabled channels.

IVIPWRMETER_VAL_CALIBRATION_IN_PROGRESS (0)

- The power meter is still taking a calibration on one or more enabled channels.

IVIPWRMETER_VAL_CALIBRATION_STATUS_UNKNOWN (-1)

- The power meter cannot determine the status of the calibration.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Is Measurement Complete

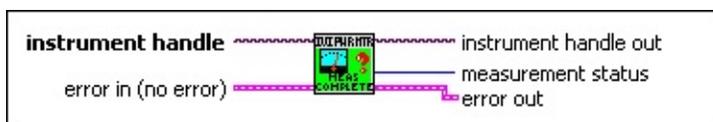
IviPwrMeterBase Capability Group

This VI queries the instrument to determine the status of the measurement initiated by the . This VI returns the IVIPWRMETER_VAL_MEAS_COMPLETE (1) value in the Status parameter only when measurements are complete on all enabled channels.

If some measurements are still in progress on one or more channels, the driver returns the IVIPWRMETER_VAL_MEAS_IN_PROGRESS (0) value. If the driver cannot query the instrument to determine its state, the driver returns the IVIPWRMETER_VAL_MEAS_STATUS_UNKNOWN (-1) value.

Note:

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **measurement status** Returns the status of the measurement.

This driver defines the following measurement status:

IVIPWRMETER_VAL_MEAS_COMPLETE (1)

- The power meter has completed the measurement on all enabled channels.

IVIPWRMETER_VAL_MEAS_IN_PROGRESS (0)

- The power meter is still taking a measurement on one or more enabled channels.

IVIPWRMETER_VAL_MEAS_STATUS_UNKNOWN (-1)

- The power meter cannot determine the status of the measurement.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Is Zero Correction Complete [ZC]

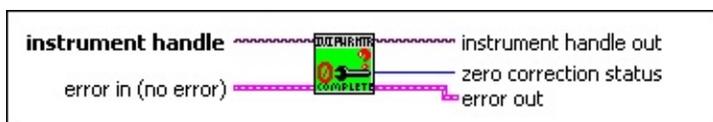
IviPwrMeterZeroCorrection Capability Group

This VI queries the instrument to determine the status of all zero correction operations initiated by the or VIs. This VI returns the `IVIPWRMETER_VAL_ZERO_COMPLETE` (1) value in the Status parameter only when zero corrections are complete on all enabled channels.

If some zero correction operations are still in progress on one or more channels, this VI returns the `IVIPWRMETER_VAL_ZERO_IN_PROGRESS` (0) value. If the driver cannot query the instrument to determine its state, the driver returns the `IVIPWRMETER_VAL_ZERO_STATUS_UNKNOWN` (-1) value.

Note:

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **zero correction status** Returns the status of the zero correction.

This driver defines the following zero correction status:

IVIPWRMETER_VAL_ZERO_COMPLETE (1)

- The power meter has completed all zero correction operations.

IVIPWRMETER_VAL_ZERO_IN_PROGRESS (0)

- The power meter is still performing a zero correction on one or more enabled channels.

IVIPWRMETER_VAL_ZERO_STATUS_UNKNOWN (-1)

- The power meter cannot determine the status of the zero correction.

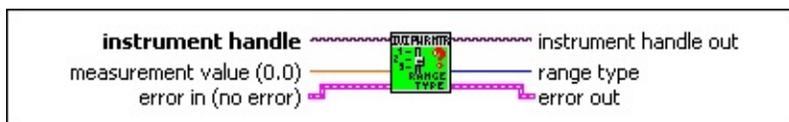
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Query Result Range Type

IviPwrMeterBase Capability Group

This VI takes a measurement value that is returned from one of the Fetch, Fetch Channel, Read, or Read Channel VIs and determines if the value is a valid measurement value or a value indicating that an out-of-range condition occurred.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

DBL **measurement value** Pass the measurement value that you obtain from one of the Fetch or Read VIs. The driver tests this value to determine if the value is a valid measurement value or a value indicating an out-of-range condition occurred.

Default Value: 0.0

ERR **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

I32 **range type** Returns whether the measurement value is a valid measurement or a value indicating that the power meter encountered an out-of-range condition.

This driver defines the following range type:

IVIPWRMETER_VAL_IN_RANGE (0)

- The measurement is within the current range limits.

IVIPWRMETER_VAL_UNDER_RANGE (-1)

- The measurement is below the current lower range limits.

IVIPWRMETER_VAL_OVER_RANGE (1)

- The measurement is above the current upper range limits.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

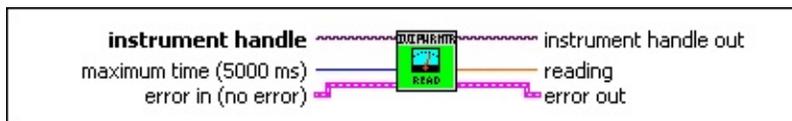
IviPwrMeter Read

IviPwrMeterBase Capability Group

This VI initiates a measurement, waits until the power meter has returned to the Idle state, and returns the result of the measurement.

After this VI executes, the value of the Reading parameter depends on the math operation specified in the .

If an out of range condition occurs on one or more enabled channels, the reading is a value indicating that an out of range condition occurred. In such a case, Reading contains an IEEE defined -Inf (Negative Infinity) or +Inf (Positive Infinity) value and the VI returns the Under Range (0x3FFA2001) or Over Range (0x3FFA2002) warning. Test if the measurement value is out of range with the .



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

I32 **maximum time (ms)** Pass the maximum length of time in which to allow the read operation to complete. Express this value in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIPWRMETER_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2020) error code. When this occurs, you can call to cancel the read operation and return the instrument to the Idle state.

Defined Values:

`IVIPWRMETER_VAL_MAX_TIME_INFINITE` (-1) - Wait indefinitely for a timeout.

`IVIPWRMETER_VAL_MAX_TIME_IMMEDIATE` (0) - Do not wait for a timeout.

Default Value: 5000

Notes

(1) The Maximum Time parameter affects only this VI. It has no effect on other timeout parameters or properties.

(2) Some specific instrument drivers do not support the Immediate Timeout or the Infinite Timeout defined values.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **reading** Returns the measured value.

For single channel measurements, Reading contains an actual reading on the channel specified by the . The unit of the result is the same as the value of the property.

For dual channel measurements, Reading contains the result of the math operation applied to the channels specified in the . The unit of the result depends on the value of the Units property and the specified operator.

For Difference and Sum operations, the resulting units is the same as the Units property.

For Quotient operations, the resulting units are in dB, except when Units are set to Watts. When set to Watts, the resulting measurement is without units.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

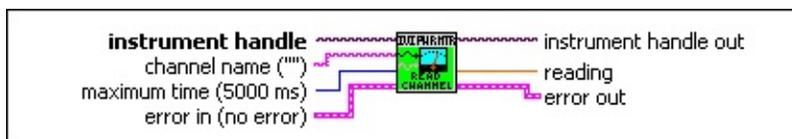
IviPwrMeter Read Channel [CH]

IviPwrMeterChannelAcquisition Capability Group

This VI initiates a measurement, waits until the power meter has returned to the Idle state, and returns the result of the measurement on the specified channel.

After this VI executes, the Reading parameter contains an actual reading on the channel specified by the Channel parameter. If the specified channel is not enabled for measurement, this VI returns the Channel Not Enabled (0xBFFA2001) error. The Reading result is in the same unit as the value of the Units property.

After this VI executes, the Reading parameter may contain a value indicating that an out-of-range condition occurred. If an out-of-range condition occurs, the Result parameter contains an IEEE defined -Inf (Negative Infinity) or +Inf (Positive Infinity) value and the VI returns the Under Range (0x3FFA2001) or Over Range (0x3FFA2002) warning. Test if the measurement value is out of range with the .



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

abc **channel name** The name of the channel from which to read the measurement.

Pass the virtual channel name that you assign to the instrument in the Action Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and

swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

 **maximum time (ms)** Pass the maximum length of time in which to allow the read operation to complete. Express this value in milliseconds.

If the operation does not complete within this time interval, the VI returns the `IVIPWRMETER_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2020) error code. When this occurs, you can call to cancel the read operation and return the instrument to the Idle state.

Defined Values:

`IVIPWRMETER_VAL_MAX_TIME_INFINITE` (-1) - Wait indefinitely for a timeout.

`IVIPWRMETER_VAL_MAX_TIME_IMMEDIATE` (0) - Do not wait for a timeout.

Default Value: 5000

Notes

(1) The Maximum Time parameter affects only this VI. It has no effect on other timeout parameters or properties.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#)

VIs.



reading The data read from the power meter.

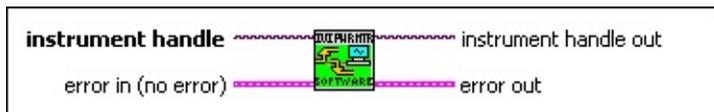


error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Send Software Trigger [SWT]

This VI sends a command to trigger the power meter. Call this VI if you pass `IVIPWRMETER_VAL_SOFTWARE_TRIG` for the property or the Trigger Source parameter of the .



Notes

- (1) If the is not set to the `IVIPWRMETER_VAL_SOFTWARE_TRIG` value, this VI returns a Trigger Not Software (0xBFFA1001) error.
- (2) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

 **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

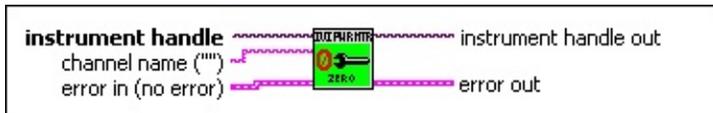
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Zero [ZC]

IviPwrMeterZeroCorrection Capability Group

This VI performs a zero correction on the specified channel. You may use the to determine when the zero correction is complete.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.
- abc** **channel name** The name for the channel for which to perform the zero correction.

Pass the virtual channel name that you assign to the instrument in the Action Utility.

Virtual channel names are aliases for instrument-specific channel strings. The instrument-specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument-specific channel through the Configuration Utility. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument-specific channel names.

Default Value: ""

- Err** **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- I/O** **instrument handle out** The instrument handle that you obtain

from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Zero All Channels [ZC]

IviPwrMeterZeroCorrection Capability Group

This VI performs a zero correction on all enabled channels. You may use the `error in (no error)` to determine when the zero correction is complete.



I/O **instrument handle** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Error In **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviPwrMeter Initialize](#) or [IviPwrMeter Initialize With Options](#) VIs.

Error Out **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviPwrMeter Properties

Expand this book to view an alphabetized list of IviPwrMeter properties.

IviPwrMeter Average Count [AVG]

IviPwrMeterAveragingCount Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	Up	IviPwrMeter Configure Averaging Count [AVG]

Description

This channel-based property specifies the average count that the instrument uses in manual averaging mode. When the [IviPwrMeter>>Basic Operation>>Averaging Auto Enabled](#) property is set to FALSE, the driver filters the input signal by averaging it the number of times specified by this property.

Property Node Path

IviPwrMeter»Averaging Count>>Average Count [AVG]

IviPwrMeter Averaging Auto Enabled

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	None	<u>IviPwrMeter Configure Averaging Auto Enabled</u>

Description

This channel-based property specifies the auto-averaging mode used by the instrument for the specified input channel.

If auto averaging is enabled, the instrument determines the best value for the averaging count automatically. The averaging count specifies the number of samples that the instrument takes before the measurement is complete.

If auto averaging is disabled, specify the averaging count explicitly by setting the [IviPwrMeter>>Averaging Count>>Average Count \[AVG\]](#) property.

Property Node Path

IviPwrMeter»Basic Operation>>Averaging Auto Enabled

Defined Values

VI_TRUE

VI_FALSE

IviPwrMeter Channel Count

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	None	None

Description

Returns the number of available channels.

Property Node Path

IviPwrMeter»Inherent IVI Settings»Specific Driver Capabilities»Channel
Count

IviPwrMeter Channel Enabled [CH]

IviPwrMeterChannelAcquisition Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	None	IviPwrMeter Configure Channel Enabled [CH]

Description

This channel-based property specifies if the power meter takes a measurement on the specified input channel. The power meter will take a measurement on a channel only if that channel is enabled.

Channels are also enabled when you call the [IviPwrMeter Configure Measurement.vi](#). See the VI description for more details.

Property Node Path

IviPwrMeter»Basic Operation>>Channel Acquisition>>Channel Enabled
[CH]

Defined Values

VI_TRUE

VI_FALSE

IviPwrMeter Correction Frequency

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	Channels	None	<u>IviPwrMeter Configure Correction Frequency</u>

Description

This channel-based property specifies the frequency of the input signal in Hertz. The instrument uses this value to determine the appropriate correction factor for the sensor.

To obtain the most accurate measurement, specify the correction frequency as close as possible to the actual frequency of the input signal.

Property Node Path

IviPwrMeter»Basic Operation>>Correction Frequency

IviPwrMeter Duty Cycle Correction Enabled [DC]

IviPwrMeterDutyCycleCorrection Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	None	IviPwrMeter Configure Duty Cycle Correction [DC]

Description

This channel-based property specifies if the power meter performs a duty cycle correction on the specified channel.

Property Node Path

IviPwrMeter»Duty Cycle>>Duty Cycle Correction Enabled [DC]

Defined Values

VI_TRUE

VI_FALSE

IviPwrMeter Duty Cycle Correction [DC]

IviPwrMeterDutyCycleCorrection Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	Up	IviPwrMeter Configure Duty Cycle Correction [DC]

Description

This channel-based property specifies the duty cycle correction the power meter uses to calculate the pulse power of a pulse-modulated signal. The power meter measures the average power of the pulsed input signal and then divides the result by the value specified for this property to obtain a pulse power reading.

The value of this property is specified as a percentage.

Property Node Path

IviPwrMeter»Duty Cycle>>Duty Cycle Correction [DC]

IviPwrMeter Internal Trigger Event Source [IT]

IviPwrMeterInternalTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviPwrMeter Configure Internal Trigger [IT]

Description

Specifies the channel that the power meter uses to monitor the internal trigger event. The power meter leaves the Idle state when the measurement signal on this channel meets the conditions set by the [IviPwrMeter>>Trigger>>Internal Trigger>>Internal Trigger Level \[IT\]](#) and the [IviPwrMeter>>Trigger>>Internal Trigger>> Internal Trigger Slope \[IT\]](#) properties.

This property affects the behavior of the instrument only if the [IviPwrMeter>>Trigger>>Trigger Source \[TRG\]](#) property is set to Internal.

Property Node Path

IviPwrMeter»Trigger>>Internal Trigger>>Internal Trigger Event Source
[IT]

IviPwrMeter Internal Trigger Level [IT]

IviPwrMeterInternalTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviPwrMeter Configure Internal Trigger Level [IT]

Description

Specifies the trigger level for the measurement signal. The power meter leaves the Idle state when the measurement signal on the channel specified by the [IviPwrMeter>>Trigger>>Internal Trigger>>Internal Trigger Event Source \[IT\]](#) property crosses the value specified by this property. The value of this property is specified in the same unit as the value of the [IviPwrMeter>>Basic Operation>>Units](#).

This property affects the behavior of the instrument only if the [IviPwrMeter>>Trigger>>Trigger Source \[TRG\]](#) property is set to Internal.

Property Node Path

IviPwrMeter»Trigger>>Internal Trigger>>Internal Trigger Level [IT]

IviPwrMeter Internal Trigger Slope [IT]

IviPwrMeterInternalTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviPwrMeter Configure Internal Trigger [IT]

Description

Specifies the polarity of the internal trigger slope. The power meter triggers on the rising or falling edge of the internal trigger source depending on the value of this property.

This property affects the behavior of the instrument only if the [IviPwrMeter>>Trigger>>Trigger Source \[TRG\]](#) property is set to Internal.

Property Node Path

IviPwrMeter»Trigger>>Internal Trigger>>Internal Trigger Slope [IT]

Defined Values

IVIPWRMETER_VAL_POSITIVE

IVIPWRMETER_VAL_NEGATIVE

IviPwrMeter Offset

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	Channels	None	None

Description

This channel-based property specifies an offset to be added to the measured value on a channel in units of dB. This property can be used to compensate for system losses or gains between the unit under test and the sensor of the power meter.

A positive value is used for loss compensation. A negative value is used for gain compensation. For example, a cable loss of 2 dB could be compensated for by setting this property to +2. Similarly, a gain stage of 10 dB could be accounted for by setting the value of this property to -10. In both cases, the reading from the power meter will indicate the power at the unit under test rather than power at the power meter's sensor.

Property Node Path

IviPwrMeter»Basic Operation>>Offset

IviPwrMeter Range Auto Enabled

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Channels	None	<u>IviPwrMeter Configure Range Auto Enabled</u>

Description

This channel-based property specifies if the power meter should automatically determine the best range for the measurement. If this property is set to TRUE, the instrument automatically determines the best range for the measurement. If this property is set to FALSE, specify the lower and upper limits of the measurement range by explicitly setting the [IviPwrMeter>>Manual Range>>Range Lower \[RNG\]](#) and [IviPwrMeter>>Manual Range>>Range Upper \[RNG\]](#) properties.

Property Node Path

IviPwrMeter»Basic Operation>>Range Auto Enabled

Defined Values

VI_TRUE

VI_FALSE

IviPwrMeter Range Lower [RNG]

IviPwrMeterManualRange Property

Data type Access Applies to Coercion High Level VIs



R/W

Channels

Down

[IviPwrMeter Configure Range \[RNG\]](#)

Description

This channel-based property specifies the lower limit (minimum) of the expected value of the measurement. The specific driver coerces this value to the appropriate range for the instrument. The value of this property is specified in the same units as the value of the [IviPwrMeter>>Basic Operation>>Units](#) property.

This property affects the behavior of the instrument only when the [IviPwrMeter>>Basic Operation>>Range Auto Enabled](#) property is set to FALSE.

Property Node Path

IviPwrMeter»Manual Range>>Range Lower [RNG]

IviPwrMeter Range Upper [RNG]

IviPwrMeterManualRange Property

Data type Access Applies to Coercion High Level VIs



R/W Channels Up

[IviPwrMeter Configure Range \[RNG\]](#)

Description

This channel-based property specifies the upper limit (maximum) of the expected value of the measurement. The specific driver coerces this value to the appropriate range for the instrument. The value of this property is specified in the same units as the value of the [IviPwrMeter>>Basic Operation>>Units](#).

This property affects the behavior of the instrument only when the [IviPwrMeter>>Basic Operation>>Range Auto Enabled](#) property is set to FALSE.

Property Node Path

IviPwrMeter»Manual Range>>Range Upper [RNG]

IviPwrMeter Ref Oscillator Enabled [RO]

IviPwrMeterReferenceOscillator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviPwrMeter Configure Reference Oscillator Enabled [RO]

Description

Specifies if the internal reference oscillator is enabled.

Property Node Path

IviPwrMeter»Reference Oscillator>>Ref Oscillator Enabled [RO]

Defined Values

VI_TRUE

VI_FALSE

IviPwrMeter Ref Oscillator Frequency [RO]

IviPwrMeterReferenceOscillator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviPwrMeter Configure Reference Oscillator [RO]

Description

Specifies the frequency of the signal generated by the reference oscillator in Hertz. This property affects the behavior of the instrument only if the [IviPwrMeter>>Reference Oscillator>>Ref Oscillator Enabled \[RO\]](#) property is set to TRUE.

Property Node Path

IviPwrMeter»Reference Oscillator>>Ref Oscillator Frequency [RO]

IviPwrMeter Ref Oscillator Level [RO]

IviPwrMeterReferenceOscillator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviPwrMeter Configure Reference Oscillator [RO]

Description

Specifies the power level of the signal generated by the reference oscillator in dBm. This property affects the behavior of the instrument only if the [IviPwrMeter>>Reference Oscillator>>Ref Oscillator Enabled \[RO\]](#) property is set to TRUE.

Property Node Path

IviPwrMeter»Reference Oscillator>>Ref Oscillator Level [RO]

IviPwrMeter Trigger Source [TRG]

IviPwrMeterTriggerSource Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviPwrMeter Configure Trigger Source [TRG]

Description

Specifies the trigger source the power meter monitors for the trigger event.

When the trigger event occurs on the source specified by this property, the power meter leaves the Wait-For-Trigger state and takes a measurement on all enabled channels.

If this property is set to the Internal defined value, the power meter uses the channel specified by the [IviPwrMeter>>Trigger>>Internal Trigger>>Internal Trigger Event Source \[IT\]](#) property to monitor the internal trigger event.

Property Node Path

IviPwrMeter»Trigger>>Trigger Source [TRG]

Defined Values

IVIPWRMETER_VAL_IMMEDIATE

IVIPWRMETER_VAL_EXTERNAL

IVIPWRMETER_VAL_INTERNAL

IVIPWRMETER_VAL_SOFTWARE_TRIG

IVIPWRMETER_VAL_TTL0

IVIPWRMETER_VAL_TTL1

IVIPWRMETER_VAL_TTL2

IVIPWRMETER_VAL_TTL3

IVIPWRMETER_VAL_TTL4

IVIPWRMETER_VAL_TTL5

IVIPWRMETER_VAL_TTL6

IVIPWRMETER_VAL_TTL7

IVIPWRMETER_VAL_ECL0

IVIPWRMETER_VAL_ECL1

IVIPWRMETER_VAL_PXI_STAR

IVIPWRMETER_VAL_RTSI_0

IVIPWRMETER_VAL_RTSI_1

IVIPWRMETER_VAL_RTSI_2

IVIPWRMETER_VAL_RTSI_3

IVIPWRMETER_VAL_RTSI_4

IVIPWRMETER_VAL_RTSI_5

IVIPWRMETER_VAL_RTSI_6

IviPwrMeter Units

IviPwrMeterBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviPwrMeter Configure Units

Description

Specifies the unit to which the RF power is converted after measurement. The actual RF power of the signal on a channel is always measured in Watts. The value of this property is used to determine the units in which the [IviPwrMeter>>Manual Range>>Range Upper \[RNG\]](#) and [IviPwrMeter>>Manual Range>>Range Lower \[RNG\]](#) properties are specified. The unit of the measurement result returned by the Read and Fetch VIs also depends on the value of this property.

Property Node Path

IviPwrMeter»Basic Operation>>Units

Defined Values

IVIPWRMETER_VAL_DBM

IVIPWRMETER_VAL_DBMV

IVIPWRMETER_VAL_DBUV

IVIPWRMETER_VAL_WATTS

IviPwrMeter Defined Values

Expand this book to view an alphabetized list of IviPwrMeter defined values.

IVIPWRMETER_VAL_CALIBRATION_COMPLETE

Description

The power meter has completed the calibration on all enabled channels.

Defined Value

1

IVIPWRMETER_VAL_CALIBRATION_IN_PROGRE

Description

The power meter is still taking a calibration on one or more enabled channels.

Defined Value

0

IVIPWRMETER_VAL_CALIBRATION_STATUS_UN

Description

The power meter cannot determine the status of the calibration.

Defined Value

-1

IVIPWRMETER_VAL_DBM

Description

Sets the units to dBm.

Defined Value

1

IVIPWRMETER_VAL_DBMV

Description

Sets the units to dB millivolts.

Defined Value

2

IVIPWRMETER_VAL_DBUV

Description

Sets the units to dB microvolts.

Defined Value

3

IVIPWRMETER_VAL_DIFFERENCE

Description

Subtract the power measured on Operand2 from the power measured on Operand1.

Defined Value

1

IVIPWRMETER_VAL_ECL0

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on ECL0.

Defined Value

200

IVIPWRMETER_VAL_ECL1

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on ECL1.

Defined Value

201

IVIPWRMETER_VAL_EXTERNAL

Description

The power meter exits the Wait-For-Trigger state when a trigger occurs on the external trigger input.

Defined Value

2

IVIPWRMETER_VAL_IMMEDIATE

Description

The power meter exits the Wait-For-Trigger state immediately after entering. It does not wait for a trigger of any kind.

Defined Value

1

IVIPWRMETER_VAL_INTERNAL

Description

The power meter exits the Wait-For-Trigger state when an internal trigger event occurs on the measurement signal.

Defined Value

3

IVIPWRMETER_VAL_MEAS_COMPLETE

Description

The power meter has completed the measurement on all enabled channels.

Defined Value

1

IVIPWRMETER_VAL_MEAS_IN_PROGRESS

Description

The power meter is still taking a measurement on one or more enabled channels.

Defined Value

0

IVIPWRMETER_VAL_MEAS_STATUS_UNKNOWN

Description

The power meter cannot determine the status of the measurement.

Defined Value

-1

IVIPWRMETER_VAL_NEGATIVE

Description

Sets the trigger event to occur on the falling edge of the trigger pulse.

Defined Value

0

IVIPWRMETER_VAL_NONE

Description

Return the measured value for Operand1.

Defined Value

0

IVIPWRMETER_VAL_POSITIVE

Description

Sets the trigger event to occur on the rising edge of the trigger pulse.

Defined Value

1

IVIPWRMETER_VAL_PXI_STAR

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on PXI Star trigger bus.

Defined Value

300

IVIPWRMETER_VAL_QUOTIENT

Description

Divide the power measured on Operand1 by the power measured on Operand2.

Defined Value

3

IVIPWRMETER_VAL_RTSI_0

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTSI0 line.

Defined Value

400

IVIPWRMETER_VAL_RTSI_1

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTS1 line.

Defined Value

401

IVIPWRMETER_VAL_RTSl_2

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTSl2 line.

Defined Value

402

IVIPWRMETER_VAL_RTSl_3

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTSl3 line.

Defined Value

403

IVIPWRMETER_VAL_RTSl_4

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTSl4 line.

Defined Value

404

IVIPWRMETER_VAL_RT5I_5

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RT5I5 line.

Defined Value

405

IVIPWRMETER_VAL_RTSl_6

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on the RTSl6 line.

Defined Value

406

IVIPWRMETER_VAL_SOFTWARE_TRIG

Description

The power meter exits the Wait-For-Trigger state when it receives a software trigger.

Defined Value

4

IVIPWRMETER_VAL_SUM

Description

Add the power measured on Operand2 to the power measured on Operand1.

Defined Value

2

IVIPWRMETER_VAL_TTL0

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL0.

Defined Value

100

IVIPWRMETER_VAL_TTL1

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL1.

Defined Value

101

IVIPWRMETER_VAL_TTL2

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL2.

Defined Value

102

IVIPWRMETER_VAL_TTL3

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL3.

Defined Value

103

IVIPWRMETER_VAL_TTL4

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL4.

Defined Value

104

IVIPWRMETER_VAL_TTL5

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL5.

Defined Value

105

IVIPWRMETER_VAL_TTL6

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL6.

Defined Value

106

IVIPWRMETER_VAL_TTL7

Description

The power meter exits the Wait-For-Trigger state when it receives a trigger on TTL7.

Defined Value

107

IVIPWRMETER_VAL_WATTS

Description

Sets the units to Watts.

Defined Value

4

IVIPWRMETER_VAL_ZERO_COMPLETE

Description

The power meter has completed all zero correction operations.

Defined Value

1

IVIPWRMETER_VAL_ZERO_IN_PROGRESS

Description

The power meter is still performing a zero correction on one or more enabled channels.

Defined Value

0

IVIPWRMETER_VAL_ZERO_STATUS_UNKNOWN

Description

The power meter cannot determine the status of the zero correction.

Defined Value

-1



IviRFSigGen Class Driver Reference for LabVIEW

IviRFSigGen VI and Property Reference

[VI Tree](#)

[Properties by Group](#)

[Error and Completion Codes](#)

IviRFSigGen Class Driver Overview

This instrument driver provides programming support for the IviRFSigGen Class. The IviRFSigGen class is designed to support the typical RF signal generator as well as common extended functionality found in more complex instruments. The IviRFSigGen class conceptualizes a RF signal generator as an instrument that can generate a sinusoidal carrier waveform with a wide variety of modulation types. The driver contains all the VIs required in the IviRFSigGen specification defined by the IVI Foundation. This driver requires NI-VISA and the IVI Compliance Package to be installed.

The IviRFSigGen class is divided into a base capability group and several extension groups. The base capability group is used to configure the carrier waveform. This includes setting the output frequency and amplitude, enabling or disabling the ALC and the RF output.

IviRFSigGen Extension Capability Groups

The IviRFSigGen class driver identifies VIs and that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name

Description

Analog and Pulse Modulation Extension Groups

[IviRFSigGenModulateAM](#)

Supports signal generators that can apply amplitude modulation to the RF output signal. You can enable or disable amplitude modulation, specify the source and coupling of the modulating signal and the modulation depth with linear or logarithmic attenuation.

[IviRFSigGenModulateFM](#)

Supports signal generators that can apply frequency modulation to the RF output signal. You can enable or disable frequency modulation, specify the source and coupling of the modulating signal and the peak frequency deviation.

[IviRFSigGenModulatePM](#)

Supports signal generators that can apply phase modulation to the RF output signal. You can enable or disable phase modulation, specify the source and coupling of the modulating signal and the peak phase deviation.

[IviRFSigGenAnalogModulationSource](#)

The source of the modulating signal is a repeated capability. One or

more internal sources (LF Generators) and/or one and more external sources can be combined. The voltage of all signals is summed before modulating the RF Signal.

IviRFSigGenModulatePulse

Supports signal generators that can apply pulse modulation to the RF output signal. You can enable or disable pulse modulation, and specify the source and the polarity of the modulating signal.

IviRFSigGenLFGenerator

Supports the LF generator (within the RF signal generator), that is normally used as a source for the modulators. To use the LF generator as a modulation source, the modulation source in the ModulateAM extensions group, ModulateFM extension group or ModulatePM extension group should be configured. The user can set the frequency and the waveform of the LF signal.

IviRFSigGenLFGeneratorOutput

Supports the output of the LF generator, if it is used as a source for external devices. You can set the amplitude and enable or disable the output. Requires the LFGenerator extension group.

IviRFSigGenPulseGenerator

Supports the pulse generator with the signal generator that is normally used as a source for the pulse modulator. You can set the pulse period, width and delay. The pulse can be triggered, so source and external trigger polarity can be set

Double pulse can be set with the functions and attributes of the IviRFSigGenPulseDoubleGenerator extension group.

The output of the pulse generator can be set with the functions and attributes of the IviRFSigGenPulseOutput extension group.

IviRFSigGenPulseDoubleGenerator

Extends the IviRFSigGenPulseGenerator extension group to support double pulse generation.

IviRFSigGenPulseGeneratorOutput

Requires the IviRFSigGenPulseGenerator extension group. The output of the pulse generator can be enabled or disabled and its polarity set.

Sweep Extension Groups

IviRFSigGenSweep

Supports signal generators with the ability to sweep (or step) the frequency or the power of the RF output signal.

IviRFSigGenFrequencySweep

Supports signal generators that can apply a frequency sweep to the output signal. You may configure the sweep with start and stop, or center and span frequencies. The sweep time is also configurable. In order to support the IviRFSigGenFrequencySweep extension group, a driver shall first support the IviRFSigGenSweep extension group. This extension group is active when the Sweep Mode attribute is set to Frequency

IviRFSigGenPowerSweep

Sweep.

Supports signal generators that can apply a power sweep to the output signal. You may configure the sweep with start and stop power. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviRFSigGenSweep extension group. This extension group is active when the Sweep Mode attribute is set to Power Sweep.

IviRFSigGenFrequencyStep

Supports signal generators that can vary (sweep) the frequency of the RF output signal in steps. You can specify the start, stop and step frequency and set linear or logarithmic spacing. Setting single step and dwell time are also included. This extension group requires the IviRFSigGenSweep extension group. Frequency stepping is enabled by setting the Sweep Mode to Frequency Step in the IviRFSigGenSweep extension group.

IviRFSigGenPowerStep

Supports signal generators that can vary (sweep) the power of the RF output signal in steps. The user can enable or disable stepping, specify the start, stop and step power. Setting single step and dwell time are also included.

This extension group requires the Sweep extension group. Power stepping is enabled by setting the Sweep Mode to Power Step in the

IviRFSigGenList

IviRFSigGenSweep extension group.

Supports signal generators that can set the frequency and power of the RF output signal to values given in a list of values. You can enable or disable stepping the frequency and power list, specify the name of the list and set its values. The active list can be selected using the list name. Setting single step and dwell time are also included.

This extension group requires the Sweep extension group. List stepping is enabled by setting the Sweep Mode to List in the IviRFSigGenSweep extension group.

Digital Modulation Extension Groups

IviRFSigGenModulateIQ

Supports signal generators that can apply IQ (vector) modulation to the RF output signal. You can enable or disable IQ modulation and specify the source of the modulating signal. A calibration is executed with an event function.

IviRFSigGenImpairmentIQ

Supports signal generators that can apply IQ (vector) modulation to the RF output signal. You can enable or disable IQ modulation and specify the source of the modulating signal. A calibration is executed with an event function.

IviRFSigGenArbGenerator

Controls the internal arbitrary waveform generator. The two

outputs of the Arb generator are used with the IQ (vector) modulation to simulate digital modulation. This group is an extension of the IviRFSigGenModulateIQ extension group.

[IviRFSigGenDigitalModulationBase](#)

Using this extension group, you can generate signals conforming to wireless communication standards (e.g. mobile cellular standards). The generated signals do not have TDMA framing nor CDMA channel coding. The functionality covers basic modulation properties such as IQ constellation, symbol mapping, etc. within a specified communication standard.

[IviRFSigGenCDMABase](#)

Use this extension group to simulate mobile (cell) phone modulation based on Code Division Multiple Access (CDMA) standard

[IviRFSigGenTDMABase](#)

Use this extension group to simulate mobile (cell) phone modulation based on Time Division Multiple Access (TDMA) standard:

Remaining Extension Groups

[IviRFSigGenALC](#)

For generators with configurable Automatic Level Control.

[IviRFSigGenReferenceOscillator](#)

Supports signal generators with a configurable frequency reference.

[IviRFSigGenSoftwareTrigger](#)

Supports RF signal generators that can generate output based on a software trigger signal. You can send a software trigger to cause signal output to occur, to start a

sweep (frequency or power), or to initiate the next step (frequency and/or power).

Related Topic

[IVI Inherent Class Capabilities](#)

IviRFSigGen Extension Capability Groups

The IviRFSigGen class driver identifies VIs and that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name

Description

[IviRFSigGenModulateAM](#)

Supports signal generators that can apply amplitude modulation to the RF output signal. You can enable or disable amplitude modulation, specify the source and coupling of the modulating signal and the modulation depth with linear or logarithmic attenuation.

[IviRFSigGenModulateFM](#)

Supports signal generators that can apply frequency modulation to the RF output signal. You can enable or disable frequency modulation, specify the source and coupling of the modulating signal and the peak frequency deviation.

[IviRFSigGenModulatePM](#)

Supports signal generators that can apply phase modulation to the RF output signal. You can enable or disable phase modulation, specify the source and coupling of the modulating signal and the peak phase deviation.

[IviRFSigGenAnalogModulationSource](#)

The source of the modulating signal is a repeated capability. One or more internal sources (LF Generators) and/or one and more external sources can be combined

The voltage of all signals is summed before modulating the RF Signal.

IviRFSigGenModulatePulse

Supports signal generators that can apply pulse modulation to the RF output signal. You can enable or disable pulse modulation, and specify the source and the polarity of the modulating signal.

IviRFSigGenLFGenerator

Supports the LF generator (within the RF signal generator), that is normally used as a source for the modulators. To use the LF generator as a modulation source, the modulation source in the ModulateAM extensions group, ModulateFM extension group or ModulatePM extension group should be configured. The user can set the frequency and the waveform of the LF signal.

IviRFSigGenLFGeneratorOutput

Supports the output of the LF generator, if it is used as a source for external devices. You can set the amplitude and enable or disable the output. Requires the LFGenerator extension group.

IviRFSigGenPulseGenerator

Supports the pulse generator with the signal generator that is normally used as a source for the pulse modulator. You can set the pulse period, width and delay. The pulse can be triggered, so source and external trigger polarity can be set. Double pulse can be set with the VIs and of the IviRFSigGenPulseDoubleGenerat

extension group.

The output of the pulse generator can be set with the VIs and of the IviRFSigGenPulseOutput extension group.

IviRFSigGenPulseDoubleGenerator

Extends the IviRFSigGenPulseGenerator extension group to support double pulse generation.

IviRFSigGenPulseGeneratorOutput

Requires the IviRFSigGenPulseGenerator extension group. The output of the pulse generator can be enabled or disabled and its polarity set.

IviRFSigGenSweep

Supports signal generators with the ability to sweep (or step) the frequency or the power of the RF output signal.

IviRFSigGenFrequencySweep

Supports signal generators that can apply a frequency sweep to the output signal. You may configure the sweep with start and stop, or center and span frequencies. The sweep time is also configurable. In order to support the IviRFSigGenFrequencySweep extension group, a driver shall first support the IviRFSigGenSweep extension group. This extension group is active when the Sweep Mode property is set to Frequency Sweep.

IviRFSigGenPowerSweep

Supports signal generators that can apply a power sweep to the output signal. You may configure the sweep with start and stop power. The sweep time is also

configurable.

In order to support this extension group, a driver shall first support the IviRFSigGenSweep extension group. This extension group is active when the Sweep Mode property is set to Power Sweep.

IviRFSigGenFrequencyStep

Supports signal generators that can vary (sweep) the frequency of the RF output signal in steps. You can specify the start, stop and step frequency and set linear or logarithmic spacing. Setting single step and dwell time are also included. This extension group requires the IviRFSigGenSweep extension group. Frequency stepping is enabled by setting the Sweep Mode to Frequency Step in the IviRFSigGenSweep extension group.

IviRFSigGenPowerStep

Supports signal generators that can vary (sweep) the power of the RF output signal in steps. The user can enable or disable stepping, specify the start, stop and step power. Setting single step and dwell time are also included.

This extension group requires the IviRFSigGenSweep extension group. Power stepping is enabled by setting the Sweep Mode attribute to Power Step in the IviRFSigGenSweep extension group.

IviRFSigGenList

Supports signal generators that can set the frequency and power of the RF output signal to values given a

a list of values. You can enable or disable stepping the frequency and power list, specify the name of the list and set its values. The active list can be selected using the list name. Setting single step and dwell time are also included.

This extension group requires the IviRFSigGenSweep extension group. List stepping is enabled by setting the Sweep Mode to List in the IviRFSigGenSweep Extension Group.

IviRFSigGenALC

For generators with configurable Automatic Level Control.

IviRFSigGenReferenceOscillator

Supports signal generators with a configurable frequency reference.

IviRFSigGenSoftwareTrigger

Supports RF signal generators that can generate output based on a software trigger signal. You can send a software trigger to cause signal output to occur, to start a sweep (frequency or power), or to initiate the next step (frequency and/or power).

This extension affects instrument behavior when the Trigger Source property is set to Software Trigger.

IviRFSigGenModulateIQ

Supports signal generators that can apply IQ (vector) modulation to the RF output signal. You can enable or disable IQ modulation and specify the source of the modulating signal. A calibration is executed with an event VI.

IviRFSigGenIQImpairment

Simulates or corrects impairment on IQ (vector) modulation. This

group is an extension of the IviRFSigGenModulateIQ Extension Group.

IviRFSigGenArbGenerator

Controls the internal arbitrary waveform generator. The two outputs of the Arb generator are used with the IQ (vector) modulation to simulate digital modulation. This group is an extension of the IviRFSigGenModulateIQ Extension Group.

IviRFSigGenDigitalModulationBase

Using this extension group, you can generate signals conforming to wireless communication standards (e.g. mobile cellular standards). The generated signals do not have TDMA framing nor CDMA channel coding. The functionality covers basic modulation properties such as IQ constellation, symbol mapping, etc. within a specified communication standard.

IviRFSigGenCDMABase

Use this extension group to simulate mobile (cell) phone modulation based on Code Division Multiple Access (CDMA) standard

IviRFSigGenTDMABase

Use this extension group to simulate mobile (cell) phone modulation based on Time Division Multiple Access (TDMA) standard:

IviRFSigGenALC Extension Group

For generators with configurable Automatic Level Control.

IviRFSigGenALC VI

[IviRFSigGen_ConfigureALC](#)

IviRFSigGenALC Properties

IVIRFSIGGEN_ATTR_ALC_SOURCE

IVIRFSIGGEN_ATTR_ALC_BANDWIDTH

IviRFSigGenAnalogModulationSource Extension Group

The source of the modulating signal is a repeated capability, which is referred to as Analog Modulation Source. One or more internal sources (LF Generators) and/or one and more external sources can be combined. The voltage of all signals is summed before modulating the RF Signal. This applies to the following extension groups:

- [IviRFSigGenModulateAM](#)
- [IviRFSigGenModulateFM](#)
- [IviRFSigGenModulatePM](#)

Typically, all the LFGenerators defined in the IviRFSigGenLFGenerator extension group have a corresponding Analog Modulation Source (or repeated capability).

IviRFSigGenAnalogModulationSource VI

[IviRFSigGen_Get_Analog_Modulation_Source_Name](#)

IviRFSigGenAnalogModulationSource Attribute

IVIRFSIGGEN_ATTR_ANALOG_MODULATION_SOURCE_COUNT

IviRFSigGenArbGenerator Extension Group

The IviRFSigGenArbGenerator extension group controls the internal arbitrary waveform generator. The two outputs of the Arb generator are used with the IQ (vector) modulation to simulate digital modulation. So this group is an extension of the IviRFSigGenModulateIQ extension group.

IviRFSigGenArbGenerator VIs

[IviRFSigGen_Configure_Arb](#)

[IviRFSigGen_Write_Arb_Waveform](#)

[IviRFSigGen_Select_Arb_Waveform](#)

[IviRFSigGen_Clear_All_Arb_Waveforms](#)

[IviRFSigGen_Query_Arb_Waveform_Capabilities](#)

[IviRFSigGen_Configure_Arb_Trigger_Source](#)

[IviRFSigGen_Configure_Arb_External_Trigger_Slope](#)

IviRFSigGenArbGenerator Properties

IVIRFSIGGEN ATTR ARB_SELECTED_WAVEFORM

IVIRFSIGGEN ATTR ARB_CLOCK_FREQUENCY

IVIRFSIGGEN ATTR ARB_FILTER_FREQUENCY

IVIRFSIGGEN ATTR ARB_MAX_NUMBER_WAVEFORMS

IVIRFSIGGEN ATTR ARB_WAVEFORM_QUANTUM

IVIRFSIGGEN ATTR ARB_WAVEFORM_SIZE_MIN

IVIRFSIGGEN ATTR ARB_WAVEFORM_SIZE_MAX

IVIRFSIGGEN ATTR ARB_TRIGGER_SOURCE

IVIRFSIGGEN ATTR ARB_EXTERNAL_TRIGGER_SLOPE

IviRFSigGen Base Capability Group

The IviRFSigGenBase capabilities group supports the basic signal generator capabilities.

IviRFSigGen VIs

[IviRFSigGen_ConfigureRF](#)

[IviRFSigGen_ConfigureALCEnabled](#)

[IviRFSigGen_ConfigureOutputEnabled](#)

[IviRFSigGen_DisableAllModulation](#)

[IviRFSigGen_WaitUntilSettled](#)

[IviRFSigGen_IsSettled](#)

IviRFSigGen Properties

[IVIRFSIGGEN_ATTR_FREQUENCY](#)

[IVIRFSIGGEN_ATTR_POWER_LEVEL](#)

[IVIRFSIGGEN_ATTR_ALC_ENABLED](#)

[IVIRFSIGGEN_ATTR_OUTPUT_ENABLED](#)

Related Topics

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

IviRFSigGenCDMABase Extension Group

With the IviRFSigGenCDMABase extension group you can simulate mobile (cell) phone modulation based on Code Division Multiple Access (CDMA) standards.

IviRFSigGenTDMABase VIs

[IviRFSigGen_SelectCDMAStandard](#)

[viRFSigGen_ConfigureCDMAClockSource](#)

[IviRFSigGen_ConfigureCDMATriggerSource](#)

[IviRFSigGen_ConfigureCDMAExternalTriggerSlope](#)

[IviRFSigGen_GetCDMATestModelName](#)

[IviRFSigGen_SelectCDMATestModel](#)

IvIRFSigGenCDMABase Properties

IVIRFSIGGEN_ATTR_CDMA_STANDARD_COUNT

VIRFSIGGEN_ATTR_CDMA_SELECTED_STANDARD

IVIRFSIGGEN_ATTR_CDMA_TRIGGER_SOURCE

IVIRFSIGGEN_ATTR_CDMA_EXTERNAL_TRIGGER_SLOPE

IVIRFSIGGEN_ATTR_CDMA_TEST_MODEL_COUNT

IVIRFSIGGEN_ATTR_CDMA_SELECTED_TEST_MODEL

IVIRFSIGGEN_ATTR_CDMA_CLOCK_SOURCE

IviRFSigGenDigitalModulationBase Extension Group

With the IviRFSigGenDigitalModulationBase extension group you can generate signals conforming to wireless communication standards (e.g. mobile cellular standards). The generated signals do not have TDMA framing nor CDMA channel coding.

The functionality covers basic modulation properties such as IQ constellation, symbol mapping, etc. within a specified communication standard.

IviRFSigGenDigitalModulationBase VIs

[IviRFSigGen_GetDigitalModulationBaseStandardName](#)

[IviRFSigGen_SelectDigitalModulationBaseStandard](#)

[IviRFSigGen_Configure_DigitalModulationBase_Clock_Source](#)

[IviRFSigGen_ConfigureDigitalModulationBaseDataSource](#)

[IviRFSigGen_ConfigureDigitalModulationBasePRBSType](#)

[IviRFSigGen_WriteDigitalModulationBaseBitSequence](#)

[IviRFSigGen_SelectDigitalModulationBaseBitSequence](#)

[IviRFSigGen_ClearAllDigitalModulationBaseBitSequences](#)

IviRFSigGenDigitalModulationBase Properties

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE STANDARD COU

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE SELECTED STAN

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE DATA SOURCE

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE PRBS TYPE

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE SELECTED BIT S

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE CLOCK SOURCE

IVIRFSIGGEN ATTR DIGITAL MODULATION BASE EXTERNAL CLOC

IviRFSigGenFrequencyStep Extension Group

The IviRFSigGenFrequencyStep Extension Group supports signal generators that can vary (sweep) the frequency of the RF output signal in steps. You can specify the start, stop and step frequency and set linear or logarithmic spacing. Setting single step and dwell time are also included.

This extension group requires the [IviRFSigGenSweep](#) extension group. Frequency stepping is enabled by setting the [Sweep Mode](#) to Frequency Step in the [IviRFSigGenSweep](#) extension group.

IviRFSigGenFrequencyStep VIs

[IviRFSigGen_Configure_Frequency_Step_Start_Stop](#)

[IviRFSigGen_ConfigureFrequencyStepDwell](#)

[IviRFSigGen_ResetFrequencyStep](#)

IviRFSigGenFrequencyStep Properties

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_START

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_STOP

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SCALING

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SIZE

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_DWELL

IviRFSigGenFrequencySweep Extension Group

The IviRFSigGenFrequencySweep extension group supports signal generators that can apply a frequency sweep to the output signal. You may configure the sweep with start and stop, or center and span frequencies. The sweep time is also configurable.

In order to support the IviRFSigGenFrequencySweep extension group, a driver shall first support the [IviRFSigGenSweep](#) extension group. The IviRFSigGenSweep extension group is active when the [Sweep Mode](#) property is set to Frequency Sweep.

IviRFSigGenFrequencySweep VIs

[IviRFSigGen_ConfigureFrequencySweepStartStop](#)

[IviRFSigGen_ConfigureFrequencySweepCenterSpan](#)

[IviRFSigGen_ConfigureFrequencySweepTime](#)

IviRFSigGenFrequencySweep Properties

IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_START

IVIRFSIGGEN_ATTR_FREQUENCY_STEP_STOP

IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_TIME

IviRFSigGenIQImpairment Extension Group

With the IviRFSigGenIQImpairment extension group you can simulate or correct impairment on IQ (vector) modulation. The IviRFSigGenIQImpairment extension group is an extension of the IviRFSigGenModulateIQ extension group.

IviRFSigGenIQImpairment VIs

[IviRFSigGen_ConfigureIQImpairmentEnabled](#)

[IviRFSigGen_ConfigureIQImpairment](#)

IvIRFSigGenIQImpairment Properties

IVIRFSIGGEN_ATTR_IQ_IMPAIRMENT_ENABLED

IVIRFSIGGEN_ATTR_IQ_I_OFFSET

IVIRFSIGGEN_ATTR_IQ_Q_OFFSET

IVIRFSIGGEN_ATTR_IQ_RATIO

IVIRFSIGGEN_ATTR_IQ_SKEW

IviRFSigGen LFGenerator Extension Group

The IviRFSigGenLFGenerator extension group supports the LF generator (within the RF signal generator), that is normally used as a source for the modulators. To use the LF generator as a modulation source, the modulation source in the [IviRFSigGenModulateAM](#) extensions group, [IviRFSigGenModulateFM](#) extension group or [IviRFSigGenModulatePM](#) extension group should be configured. You can set the frequency and the waveform of the LF signal.

IviRFSigGenLFGenerator VIs

[IviRFSigGen_GetLFGeneratorName](#)

[IviRFSigGen_SetActiveLFGenerator](#)

[IviRFSigGen_ConfigureLFGeneratorOutput](#)

IviRFSigGenLFGenerator Properties

IVIRFSIGGEN_ATTR_ACTIVE_LF_GENERATOR

IVIRFSIGGEN_ATTR_LF_GENERATOR_COUNT

IVIRFSIGGEN_ATTR_LF_GENERATOR_FREQUENCY

IVIRFSIGGEN_ATTR_LF_GENERATOR_WAVEFORM

IviRFSigGenLFGeneratorOutput Extension Group

The IviRFSigGenLFGenerator Output extension group supports the output of the LF generator, if it is used as a source for external devices. You can set the amplitude and enable or disable the output. The IviRFSigGenLFGenerator Output requires the [IviRFSigGenLFGenerator](#) extension group.

IviRFSigGenLFGeneratorOutput VI

[IviRFSigGen_ConfigureLFGeneratorOutput](#)

IviRFSigGenLFGeneratorOutput Properties

IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_AMPLITUDE

IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_ENABLED

IviRFSigGenList Extension Group

The IviRFSigGenList extension group supports signal generators that can set the frequency and power of the RF output signal to values given as a list of values. You can enable or disable stepping the frequency and power list, specify the name of the list and set its values. The active list can be selected using the list name. Setting single step and dwell time are also included.

The IviRFSigGenList extension group requires the [IviRFSigGenSweep](#) extension group. List stepping is enabled by setting the [Sweep Mode](#) attribute to List in the [IviRFSigGenSweep](#) extension group.

IviRFSigGenList VIs

[IviRFSigGen_CreateFrequencyList](#)

[IviRFSigGen_CreatePowerList](#)

[IviRFSigGen_CreateFrequencyPowerList](#)

[IviRFSigGen_SelectList](#)

[IviRFSigGen_ClearAllLists](#)

[IviRFSigGen_ConfigureListDwell](#)

[IviRFSigGen_ResetList](#)

IviRFSigGenList Properties

IVIRFSIGGEN_ATTR_LIST_SELECTED_NAME

IVIRFSIGGEN_ATTR_LIST_SINGLE_STEP_ENABLED

IVIRFSIGGEN_ATTR_LIST_DWELL

IviRFSigGenModulateAM Extension Group

The IviRFSigGenModulateAM extension group supports signal generators that can apply amplitude modulation to the RF output signal. You can enable or disable amplitude modulation, specify the source and coupling of the modulating signal and the modulation depth with linear or logarithmic attenuation.

IviRFSigGenModulateAM VIs

[IviRFSigGen_ConfigureAMEnabled](#)

[IviRFSigGen_ConfigureAMExternalCoupling](#)

[IviRFSigGen_ConfigureAM](#)

IviRFSigGenModulateAM Properties

IVIRFSIGGEN ATTR AM ENABLED

IVIRFSIGGEN ATTR AM SOURCE

IVIRFSIGGEN ATTR AM SCALING

IVIRFSIGGEN ATTR AM EXTERNAL COUPLING

IVIRFSIGGEN ATTR AM NOMINAL VOLTAGE

IVIRFSIGGEN ATTR AM DEPTH

IviRFSigGenModulateFM Extension Group

The IviRFSigGenModulateFM extension group supports signal generators that can apply frequency modulation to the RF output signal. You can enable or disable frequency modulation, specify the source and coupling of the modulating signal and the peak frequency deviation.

IviRFSigGenModulateFM VIs

[IviRFSigGen_ConfigureFMEnabled](#)

[IviRFSigGen_ConfigureFMExternalCoupling](#)

[IviRFSigGen_ConfigureFM](#)

IviRFSigGenModulateFM Properties

IVIRFSIGGEN_ATTR_FM_ENABLED

IVIRFSIGGEN_ATTR_FM_SOURCE

IVIRFSIGGEN_ATTR_FM_EXTERNAL_COUPLING

IVIRFSIGGEN_ATTR_FM_NOMINAL_VOLTAGE

IVIRFSIGGEN_ATTR_FM_DEVIATION

IviRFSigGenModulateIQ Extension Group

The IviRFSigGenModulateIQ extension group supports signal generators that can apply IQ (vector) modulation to the RF output signal. The user can enable or disable IQ modulation and specify the source of the modulating signal. A calibration is executed with an event VI.

IviRFSigGenModulateIQ VIs

[IviRFSigGen_ConfigureIQEnabled](#)

[IviRFSigGen_ConfigureIQ](#)

[IviRFSigGen_CalibrateIQ](#)

IviRFSigGenModulateIQ Properties

IVIRFSIGGEN_ATTR_IQ_ENABLED

IVIRFSIGGEN_ATTR_IQ_SOURCE

IVIRFSIGGEN_ATTR_IQ_NOMINAL_VOLTAGE

IVIRFSIGGEN_ATTR_IQ_SWAP_ENABLED

IviRFSigGenModulatePM Extension Group

The IviRFSigGenModulatePM extension group supports signal generators that can apply phase modulation to the RF output signal. You can enable or disable phase modulation, specify the source and coupling of the modulating signal and the peak phase deviation.

IviRFSigGenModulatePM VIs

[IviRFSigGen_ConfigurePMEnabled](#)

[IviRFSigGen_ConfigurePMExternalCoupling](#)

[IviRFSigGen_ConfigurePM](#)

IviRFSigGenModulatePM Properties

IVIRFSIGGEN_ATTR_PM_ENABLED

IVIRFSIGGEN_ATTR_PM_SOURCE

IVIRFSIGGEN_ATTR_PM_EXTERNAL_COUPLING

IVIRFSIGGEN_ATTR_PM_NOMINAL_VOLTAGE

IVIRFSIGGEN_ATTR_PM_DEVIATION

IviRFSigGenModulatePulse Extension Group

The IviRFSigGenModulatePulse extension group supports signal generators that can apply pulse modulation to the RF output signal. You can enable or disable pulse modulation, and specify the source and the polarity of the modulating signal.

IviRFSigGenModulatePulse VIs

[IviRFSigGen_ConfigurePulseModulationEnabled](#)

[IviRFSigGen_ConfigurePulseModulationSource](#)

[IviRFSigGen_ConfigurePulseModulationExternalPolarity](#)

IviRFSigGenModulatePulse Properties

IVIRFSIGGEN_ATTR_PULSE_MODULATION_ENABLED

IVIRFSIGGEN_ATTR_PULSE_MODULATION_SOURCE

IVIRFSIGGEN_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY

IviRFSigGenPowerStep Extension Group

The IviRFSigGenPowerStep extension group supports signal generators that can vary (sweep) the power of the RF output signal in steps. You can enable or disable stepping, specify the start, stop and step power. Setting single step and dwell time are also included. The IviRFSigGenPowerStep extension group requires the [IviRFSigGenSweep](#) extension group. Power stepping is enabled by setting the [Sweep Mode](#) property to Power Step in the [IviRFSigGenSweep](#) extension group.

IviRFSigGenPowerStep VIs

[IviRFSigGen_ConfigurePowerStepStartStop](#)

[IviRFSigGen_ConfigurePowerStepDwell](#)

[IviRFSigGen_ResetPowerStep](#)

IviRFSigGenPowerStep Properties

IVIRFSIGGEN_ATTR_POWER_STEP_START

IVIRFSIGGEN_ATTR_POWER_STEP_STOP

IVIRFSIGGEN_ATTR_POWER_STEP_SIZE

IVIRFSIGGEN_ATTR_POWER_STEP_SINGLE_STEP_ENABLED

IVIRFSIGGEN_ATTR_POWER_STEP_DWELL

IviRFSigGenPowerSweep Extension Group

The IviRFSigGenPowerSweep extension group supports signal generators that can apply a power sweep to the output signal. You may configure the sweep with start and stop power. The sweep time is also configurable.

In order to support the IviRFSigGenPowerSweep extension group, a driver shall first support the [IviRFSigGenSweep](#) extension group. The [IviRFSigGenSweep](#) extension group is active when the [Sweep Mode](#) property is set to Power Sweep.

IviRFSigGenPowerSweep VIs

[IviRFSigGen_ConfigurePowerSweepStartStop](#)

[IviRFSigGen_ConfigurePowerSweepTime](#)

IviRFSigGenPowerSweep Properties

IVIRFSIGGEN_ATTR_POWER_SWEEP_START

IVIRFSIGGEN_ATTR_POWER_SWEEP_STOP

IVIRFSIGGEN_ATTR_POWER_SWEEP_TIME

IviRFSigGenPulseDoubleGenerator Extension Group

The IviRFSigGenPulseDoubleGenerator extension group extends the [IviRFSigGenPulseGenerator](#) extension group to support double pulse generation.

IviRFSigGenPulseDoubleGenerator VI

[IviRFSigGen_ConfigurePulseDouble](#)

IviRFSigGenPulseDoubleGenerator Properties

IVIRFSIGGEN_ATTR_PULSE_DOUBLE_ENABLED

IVIRFSIGGEN_ATTR_PULSE_DOUBLE_DELAY

IviRFSigGenPulseGenerator Extension Group

The IviRFSigGenPulseGenerator extension group supports the pulse generator within the signal generator that is normally used as a source for the pulse modulator. You can set the pulse period, width and delay. The pulse can be triggered, so source and external trigger polarity can be set.

Double pulse can be set with the VIs and properties of the [IviRFSigGenPulseDoubleGenerator](#) extension group. The output of the pulse generator can be set with the VIs and properties of the [IviRFSigGenPulseGeneratorOutput](#) extension group.

IviRFSigGenPulseGenerator VIs

[IviRFSigGen_ConfigurePulseExternalTrigger](#)

[IviRFSigGen_ConfigurePulseInternalTrigger](#)

[IviRFSigGen_ConfigurePulse](#)

IviRFSigGenPulseGenerator Properties

IVIRFSIGGEN_ATTR_PULSE_INTERNAL_TRIGGER_PERIOD

IVIRFSIGGEN_ATTR_PULSE_WIDTH

IVIRFSIGGEN_ATTR_PULSE_GATING_ENABLED

IVIRFSIGGEN_ATTR_PULSE_TRIGGER_SOURCE

IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_SLOPE

IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_DELAY

IviRFSigGenPulseGeneratorOutput Extension Group

The IviRFSigGenPulseGeneratorOutput extension group requires the [IviRFSigGenPulseGenerator](#) extension group. The output of the pulse generator can be enabled or disabled and its polarity set.

IviRFSigGenPulseGeneratorOutput Properties

IVIRFSIGGEN_ATTR_PULSE_OUTPUT_POLARITY

IVIRFSIGGEN_ATTR_PM_SOURCE

IviRFSigGen ReferenceOscillator Extension Group

The IviRFSigGenReferenceOscillator extension group supports signal generators with a configurable frequency reference.

IviRFSigGen ReferenceOscillator VI

[IviRFSigGen_ConfigureReferenceOscillator](#)

IviRFSigGen ReferenceOscillator Properties

IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_SOURCE

IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY

IviRFSigGenSoftwareTrigger Extension Group

The IviRFSigGenSoftwareTrigger extension group supports RF signal generators that can generate output based on a software trigger signal. You can send a software trigger to cause signal output to occur, to start a sweep (frequency or power), or to initiate the next step (frequency and/or power). The IviRFSigGenSoftwareTrigger extension group affects instrument behavior when the Trigger Source property is set to Software Trigger.

IviRFSigGenSoftwareTrigger VI

[IviRFSigGen_SendSoftwareTrigger](#)

IviRFSigGenSweep Extension Group

The IviRFSigGenSweep extension group supports signal generators with the ability to sweep (or step) the frequency or the power of the RF output signal.

IviRFSigGenSweep VI

[IviRFSigGen_ConfigureSweep](#)

IviRFSigGenSweep Properties

IVIRFSIGGEN_ATTR_SWEEP_MODE

IVIRFSIGGEN_ATTR_SWEEP_TRIGGER_SOURCE

IviRFSigGenTDMABase Extension Group

With IviRFSigGenTDMABase Extension Group you can simulate mobile (cell) phone modulation based on Time Division Multiple Access (TDMA) standards.

IviRFSigGenTDMABase VIs

IviRFSigGen_GetTDMAStandardName

viRFSigGen_SelectTDMAStandard

IviRFSigGen_ConfigureTDMAClockSource

IviRFSigGen_ConfigureTDMATriggerSource

IviRFSigGen_ConfigureTDMAExternalTriggerSlope

IviRFSigGen_GetTDMAFrameName

IviRFSigGen_SelectTDMAFrame

IviRFSigGenTDMABase Properties

VIRFSIGGEN_ATTR_TDMA_SELECTED_STANDARD

IVIRFSIGGEN_ATTR_TDMA_TRIGGER_SOURCE

IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_TRIGGER_SLOPE

IVIRFSIGGEN_ATTR_TDMA_SELECTED_FRAME

IVIRFSIGGEN_ATTR_TDMA_CLOCK_SOURCE

IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_CLOCK_TYPE

IviRFSigGen Behavior Models

[IviRFSigGenALC](#)

[IviRFSigGenAnalogModulationSource](#)

[IviRFSigGenArbGenerator](#)

[IviRFSigGenBase](#)

[IviRFSigGenCDMABase](#)

[IviRFSigGenDigitalModulationBase](#)

[IviRFSigGenFrequencyStep](#)

[IviRFSigGenFrequencySweep](#)

[IviRFSigGenIQImpairment](#)

[IviRFSigGenIviRFSigGen](#)

[IviRFSigGenLFGenerator](#)

[IviRFSigGenLFGeneratorOutput](#)

[IviRFSigGenList](#)

[IviRFSigGenModulateAM](#)

[IviRFSigGenModulateFM](#)

[IviRFSigGenModulateIQ](#)

[IviRFSigGenModulatePM](#)

[IviRFSigGenModulatePulse](#)

[IviRFSigGenPowerStep](#)

[IviRFSigGenPowerSweep](#)

[IviRFSigGenPulseDoubleGenerator](#)

[IviRFSigGenPulseGenerator](#)

[IviRFSigGenPulseGeneratorOutput](#)

[IviRFSigGenReferemceOscillator](#)

[IviRFSigGenSoftwareTrigger](#)

[IviRFSigGenSweep](#)

[IviRFSigGenTDMABase](#)

IviRFSigGenALC Behavior Model

The IviRFSigGenALC extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenALC settings.

IviRFSigGenAnalogModulationSource Behavior Model

The IviRFSigGenAnalogModulationSource extension group follows the behavior model of the [IviRFSigGenBase](#) capability group.

IviRFSigGenArbGenerator Behavior Model

The IviRFSigGenArbGenerator extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenArbGenerator settings.

IviRFSigGenBase Behavior Model

After you call the [Initialize](#) or [Reset](#) VIs, the RF signal generator produces an output signal based on its current configuration. All changes to the RF signal generator's IviRFSigGenBase VIs and take place immediately. If you execute the [Wait Until Settled VI](#), the driver will block any further operation until the VI completes (such as, the signal at the output of the RF signal generator has settled).

IviRFSigGenCDMABase Behavior Model

The IviRFSigGenCDMABase extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenCDMABase settings.

IviRFSigGenDigitalModulationBase Behavior Model

The IviRFSigGenDigitalModulationBase extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenDigitalModulationBase settings.

IviRFSigGenFrequencyStep Behavior Model

The IviRFSigGenFrequencyStep extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group.

IviRFSigGenFrequencySweep Behavior Model

The IviRFSigGenFrequencySweep extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group.

IviRFSigGenIQImpairment Behavior Model

The IviRFSigGenIQImpairment extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenIQImpairment settings.

IviRFSigGen LFGenerator Behavior Model

The IviRFSigGen LFGenerator extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGen LFGenerator settings.

IviRFSigGen LFGeneratorOutput Behavior Model

The IviRFSigGen LFGeneratorOutput extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGen LFGeneratorOutput settings.

IviRFSigGenList Behavior Model

The IviRFSigGenList extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group.

IviRFSigGenModulateAM Behavior Model

The IviRFSigGenModulateAM extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenModulateAM settings.

IviRFSigGenModulateFM Behavior Model

The IviRFSigGenModulateFM extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenModulateFM settings.

IviRFSigGenModulateIQ Behavior Model

The IviRFSigGenModulateIQ extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenModulateIQ settings.

IviRFSigGenModulatePM Behavior Model

The IviRFSigGenModulatePM extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenModulatePM settings.

IviRFSigGenModulatePulse Behavior Model

The IviRFSigGenModulatePulse extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulatePulse settings.

IviRFSigGenPowerStep Behavior Model

The IviRFSigGenPowerStep extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group.

IviRFSigGenPowerSweep Behavior Model

The IviRFSigGenPowerSweep extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group.

IviRFSigGenPulseDoubleGenerator Behavior Model

The IviRFSigGenPulseDoubleGenerator extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenPulseDoubleGenerator settings.

IviRFSigGenPulseGenerator Behavior Model

The IviRFSigGenPulseGenerator extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure [IviRFSigGenPulseGenerator](#) settings.

IviRFSigGenPulseGeneratorOutput Behavior Model

The IviRFSigGenPulseGeneratorOutput extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure [IviRFSigGenPulseGeneratorOutput](#) settings.

IviRFSigGen ReferenceOscillator Behavior Model

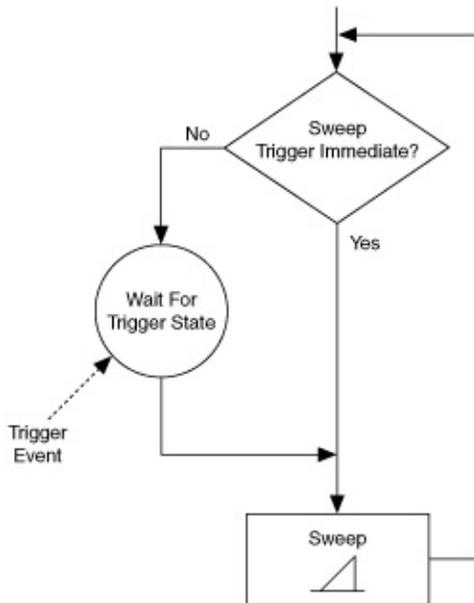
The IviRFSigGenReferenceOscillator extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenReferenceOscillator settings.

IviRFSigGen SoftwareTrigger Behavior Model

The IviRFSigGenSoftwareTrigger extension group follows the behavior model of the [IviRFSigGenSweep](#) capability group. The only modification to the behavior model from the [IviRFSigGenSweep](#) capability group is the ability to send software triggers.

The following behavior models show the relationship between the IviRFSigGenSweep extension group and RF Signal Generator behavior.

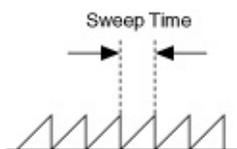
Frequency Sweep/Power Sweep



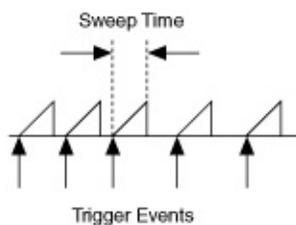
IviRFSigGenSweep Behavior Model

Setting the [Sweep Trigger Source](#) property to Immediate will continuously generate sweeps (either frequency or power sweeps). The duration of one sweep from start to stop is defined with Frequency Sweep Time or Power Sweep Time. Setting the [Sweep Trigger Source](#) property to External or Software Trigger will delay the start of the next sweep until the specified trigger event occurs.

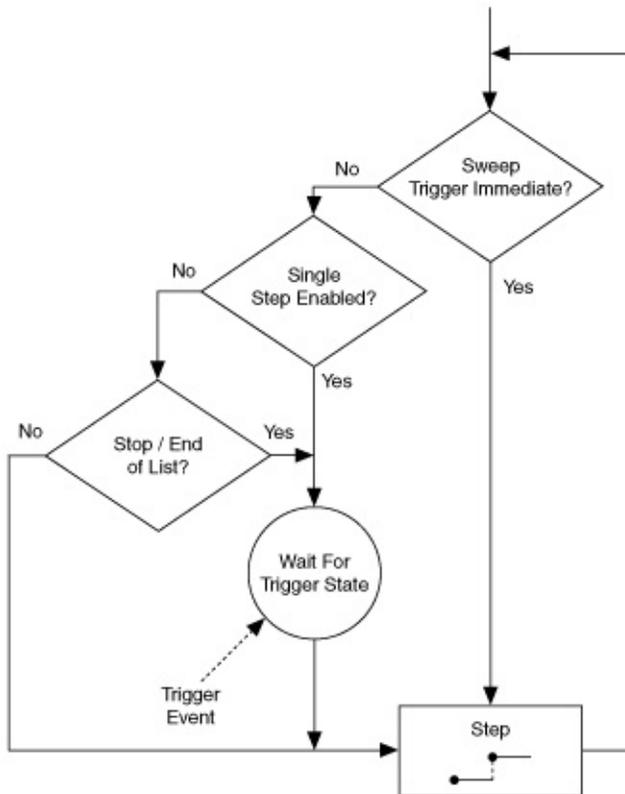
Trigger Immediate:



Trigger External/Software:

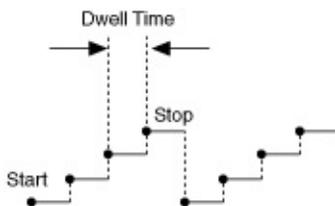


Frequency Step/Power Step/List

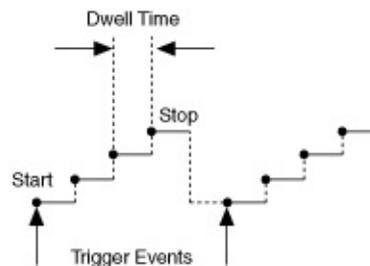


Setting the [Sweep Trigger](#) property to Immediate will continuously generate steps (frequency or power) either from start to stop with fixed increments (steps) or from a list of arbitrary values. The duration of one step is defined with Dwell Time. Setting the [Sweep Trigger](#) property to External or Software Trigger will delay the start of the next step until the specified trigger event occurs.

Trigger Immediate:

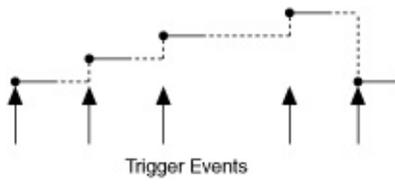


Trigger External/Software:



Single Step and Trigger

External/Software:

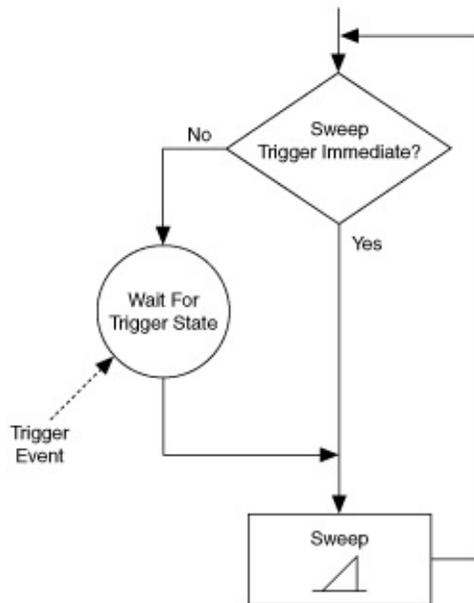


IviRFSigGenSweep Behavior Model

Setting the [Sweep Trigger Source](#) property to Immediate will continuously generate steps (frequency or power) either from start to stop with fixed increments (steps) or from a list of arbitrary values. The duration of one step is defined with Dwell Time. Setting the [Sweep Trigger Source](#) property to External or Software Trigger will delay the start of the next step until the specified trigger event occurs.

The following behavior models show the relationship between the IviRFSigGenSweep extension group and RF Signal Generator behavior.

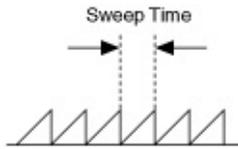
Frequency Sweep/Power Sweep



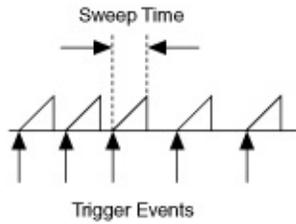
IviRFSigGenSweep Behavior Model

Setting the [Sweep Trigger Source](#) property to Immediate will continuously generate sweeps (either frequency or power sweeps). The duration of one sweep from start to stop is defined with Frequency Sweep Time or Power Sweep Time. Setting the [Sweep Trigger Source](#) property to External or Software Trigger will delay the start of the next sweep until the specified trigger event occurs.

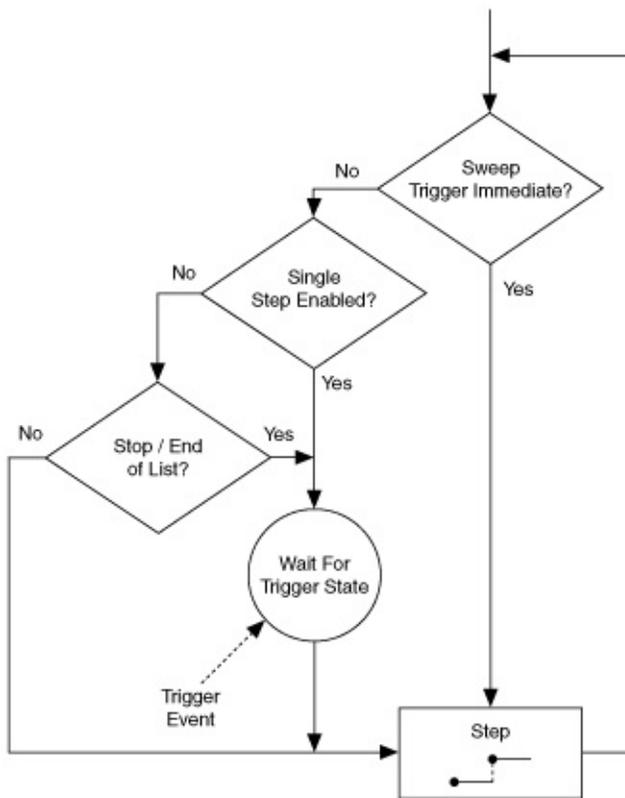
Trigger Immediate:



Trigger External/Software:



Frequency Step/Power Step/List

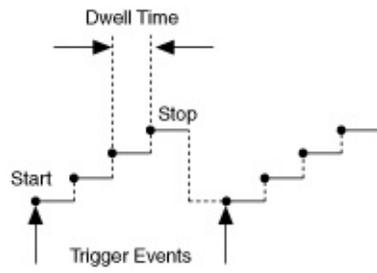
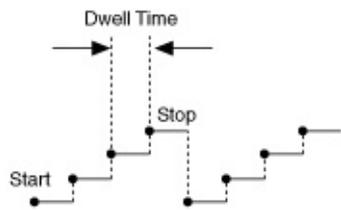


Setting the [Sweep Trigger](#) property to Immediate will continuously generate steps (frequency or power) either from start to stop with fixed increments (steps) or from a list of arbitrary values. The duration of one step is defined with Dwell Time. Setting the [Sweep Trigger](#) property to External or Software Trigger will delay the start of the next step until the specified trigger event occurs.

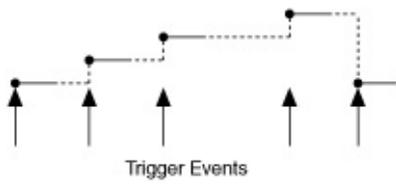
Trigger Immediate:

Trigger

External/Software:



Single Step and Trigger
External/Software:



IviRFSigGenTDMABase Behavior Model

The IviRFSigGenTDMABase extension group follows the behavior model of the [IviRFSigGenBase](#) capability group. The only modification to the behavior model from the [IviRFSigGenBase](#) capability group is the ability to configure IviRFSigGenTDMABase settings.

IviRFSigGen VI Tree

Class/Panel Name	VI Name
Configuration Functions	
RF	
Configure RF	<u>IviRFSigGen_Con RF.vi</u>
Configure ALC Enabled	<u>IviRFSigGen_Con ALC Enabled.vi</u>
Configure Output Enabled	<u>IviRFSigGen_Con Output Enabled.vi</u>
Analog Modulation	
Configure AM Enabled [AM]	<u>IviRFSigGen_Con AM Enabled [AM].</u>
Configure AM External Coupling [AM]	<u>IviRFSigGen_Con AM External Coup [AM].vi</u>
Configure AM [AM]	<u>IviRFSigGen_Con AM [AM].vi</u>
Configure FM Enabled [FM]	<u>IviRFSigGen_Con FM Enabled [FM].</u>
Configure FM External Coupling [FM]	<u>IviRFSigGen_Con FM External Coup [FM].vi</u>
Configure FM [FM]	<u>IviRFSigGen_Con FM [FM].vi</u>
Configure PM Enabled [PM]	<u>IviRFSigGen_Con PM Enabled [PM].</u>
Configure PM External Coupling [PM]	<u>IviRFSigGen_Con PM External Coup [PM].vi</u>
Configure PM [PM]	<u>IviRFSigGen_Con PM [PM].vi</u>

Get Analog Modulation Source Name [MS] [IviRFSigGen_Get Analog Modulation Source Name \[MS\]](#)

Pulse Modulation

Configure Pulse Modulation Enabled [PULM] [IviRFSigGen_Con Pulse Modulation Enabled \[PULM\].v](#)

Configure Pulse Modulation Source [PULM] [IviRFSigGen_Con Pulse Modulation Source \[PULM\].vi](#)

Configure Pulse Modulation External Polarity [PULM] [IviRFSigGen_Con Pulse Modulation External Polarity \[PULM\].vi](#)

LF Generator

Configure LF Generator [LF] [IviRFSigGen_Con LF Generator \[LF\]](#)

Get LF Generator Name [LF] [IviRFSigGen_Get Generator Name \[LF\]](#)

Set Active LF Generator [LF] [IviRFSigGen_Set Active LF Generator \[LF\].vi](#)

Configure LF Generator Output [LFO] [IviRFSigGen_Con LF Generator Output \[LFO\].vi](#)

Pulse Generator

Configure Pulse External Trigger [PG] [IviRFSigGen_Con Pulse External Trigger \[PG\].vi](#)

Configure Pulse Internal Trigger [PG] [IviRFSigGen_Con Pulse Internal Trigger \[PG\].vi](#)

Configure Pulse [PG] [IviRFSigGen_Con Pulse \[PG\].vi](#)

Configure Pulse Double [DPG]

[IviRFSigGen_Con
Pulse Double \[DPG\].vi](#)

Configure Pulse Output [PGO]

[IviRFSigGen_Con
Pulse Output \[PGO\].vi](#)

Sweep

Configure Sweep

[IviRFSigGen_Con
Sweep.vi](#)

Configure Frequency Sweep Start Stop [FSW]

[IviRFSigGen_Con
Frequency Sweep
Stop \[FSW\].vi](#)

Configure Frequency Sweep Center Span [FSW]

[IviRFSigGen_Con
Frequency Sweep
Center Span \[FSW\].vi](#)

Configure Frequency Sweep Time [FSW]

[IviRFSigGen_Con
Frequency Sweep
\[FSW\].vi](#)

Configure Power Sweep Start Stop [PSW]

[IviRFSigGen_Con
Power Sweep Sta
Stop \[PSW\].vi](#)

Configure Power Sweep Time [PSW]

[IviRFSigGen_Con
Power Sweep Tim
\[PSW\].vi](#)

Configure Frequency Step Start Stop [FST]

[IviRFSigGen_Con
Frequency Step S
Stop \[FST\].vi](#)

Configure Frequency Step Dwell [FST]

[IviRFSigGen_Con
Frequency Step D
\[FST\].vi](#)

Reset Frequency Step [FST]

[IviRFSigGen_Res
Frequency Step
\[FST\].vi](#)

Configure Power Step Start Stop [PST]

[IviRFSigGen_Con
Power Step Start :
\[PST\].vi](#)

Configure Power Step Dwell [PST]

[IviRFSigGen_Con
Power Step Dwell
\[PST\].vi](#)

Reset Power Step [PST]

[IviRFSigGen_Res
Power Step \[PST\]](#)

List

Create Frequency List [LST]

[IviRFSigGen_Cre
Frequency List \[LS](#)

Create Power List [LST]

[IviRFSigGen_Cre
Power List \[LST\].v](#)

Create Frequency Power List [LST]

[IviRFSigGen_Cre
Frequency Power
\[LST\].vi](#)

Select List [LST]

[IviRFSigGen_Sele
List \[LST\].vi](#)

Clear All Lists [LST]

[IviRFSigGen_Clea
Lists \[LST\].vi](#)

Configure List Dwell [LST]

[IviRFSigGen_Con
List Dwell \[LST\].vi](#)

Reset List [LST]

[IviRFSigGen_Res
List \[LST\].vi](#)

ALC

Configure ALC [ALC]

[IviRFSigGen_Con
ALC \[ALC\].vi](#)

Reference Oscillator

Configure Reference Oscillator [RO]

[IviRFSigGen_Con
Reference Oscillat
\[RO\].vi](#)

IQ

Configure IQ Enabled [MIQ]

[IviRFSigGen_Con
IQ Enabled \[MIQ\].](#)

Configure IQ [MIQ]

[IviRFSigGen_Con
IQ \[MIQ\].vi](#)

Calibrate IQ [MIQ] [IviRFSigGen_Cali
IQ \[MIQ\].vi](#)

Configure IQ Impairment Enabled [IQI] [IviRFSigGen_Con
IQ Impairment Ena
\[IQI\].vi](#)

Configure IQ Impairment [IQI] [IviRFSigGen_Con
IQ Impairment \[IQ](#)

ARB Generator

Configure Arb [ARB] [IviRFSigGen_Con
Arb \[ARB\].vi](#)

Write Arb Waveform [ARB] [IviRFSigGen_Writ
Waveform \[ARB\].v](#)

Select Arb Waveform [ARB] [IviRFSigGen_Sele
Arb Waveform \[AF](#)

Clear All Arb Waveforms [ARB] [IviRFSigGen_Clea
Arb Waveforms
\[ARB\].vi](#)

Query Arb Waveform Capabilities [ARB] [IviRFSigGen_Que
Arb Waveform
Capabilities \[ARB\]](#)

Configure Arb Trigger Source [ARB] [IviRFSigGen_Con
Arb Trigger Sourc
\[ARB\].vi](#)

Configure Arb External Trigger Slope [ARB] [IviRFSigGen_Con
Arb External Trigg
Slope \[ARB\].vi](#)

Digital Modulation Base

Get Digital Modulation Base Standard Name [DMB] [IviRFSigGen_Get
Digital Modulation
Standard Name
\[DMB\].vi](#)

Select DigitalModulationBase Standard [DMB] [IviRFSigGen_Sele
DigitalModulationE
Standard \[DMB\].vi](#)

Configure DigitalModulationBase Clock Source [DMB] [IviRFSigGen_Con DigitalModulationE Clock Source \[DM](#)

Configure DigitalModulationBase Data Source [DMB] [IviRFSigGen_Con DigitalModulationE Data Source \[DME](#)

Configure DigitalModulationBase PRBS Type [DMB] [IviRFSigGen_Con DigitalModulationE PRBS Type \[DMB\]](#)

Write DigitalModulationBase Bit Sequence [DMB] [IviRFSigGen_Writ DigitalModulationE Bit Sequence \[DM](#)

Select DigitalModulationBase Bit Sequence [DMB] [IviRFSigGen_Sele DigitalModulationE Bit Sequence \[DM](#)

Clear All DigitalModulationBase Bit Sequences [DMB] [IviRFSigGen_Clea DigitalModulationE Bit Sequences \[DM](#)

CDMA

Get CDMA Standard Name [CDMA] [IviRFSigGen_Get CDMA Standard N \[CDMA\].vi](#)

Select CDMA Standard [CDMA] [IviRFSigGen_Sele CDMA Standard \[CDMA\].vi](#)

Configure CDMA Clock Source [CDMA] [IviRFSigGen_Con CDMA Clock Sour \[CDMA\].vi](#)

Configure CDMA Trigger Source [CDMA] [IviRFSigGen_Con CDMA Trigger Sou \[CDMA\].vi](#)

Configure CDMA External Trigger Slope [CDMA] [IviRFSigGen_Con CDMA External Tr Slope \[CDMA\].vi](#)

Get CDMA Test Model Name [CDMA]

[IviRFSigGen_Get
CDMA Test Model
Name \[CDMA\].vi](#)

Select CDMA Test Model [CDMA]

[IviRFSigGen_Select
CDMA Test Model
\[CDMA\].vi](#)

TDMA

Get TDMA Standard Name [TDMA]

[IviRFSigGen_Get
TDMA Standard Name
\[TDMA\].vi](#)

Select TDMA Standard [TDMA]

[IviRFSigGen_Select
TDMA Standard
\[TDMA\].vi](#)

Configure TDMA Clock Source [TDMA]

[IviRFSigGen_Configure
TDMA Clock Source
\[TDMA\].vi](#)

Configure TDMA Trigger Source [TDMA]

[IviRFSigGen_Configure
TDMA Trigger Source
\[TDMA\].vi](#)

Configure TDMA External Trigger Slope [TDMA]

[IviRFSigGen_Configure
TDMA External Trigger
Slope \[TDMA\].vi](#)

Get TDMA Frame Name [TDMA]

[IviRFSigGen_Get
TDMA Frame Name
\[TDMA\].vi](#)

Select TDMA Frame [TDMA]

[IviRFSigGen_Select
TDMA Frame
\[TDMA\].vi](#)

Action Functions

Disable All Modulation

[IviRFSigGen_Disable
All Modulation.vi](#)

Wait Until Settled

[IviRFSigGen_Wait
Until Settled.vi](#)

Is Settled

[IviRFSigGen_Is
Settled.vi](#)

Send Software Trigger [SWT]

[Settled.vi](#)

[IviRFSigGen_Sen
Software Trigger
\[SWT\].vi](#)

IviRFSigGen Properties

Group/Attribute Name	Property
RF	
Frequency	IviRFSigGen»RF»F
Power Level	IviRFSigGen»RF»F
ALC Enabled	IviRFSigGen»RF»/
Output Enabled	IviRFSigGen»RF»C
Analog Modulation	
AM	
AM Enabled [AM]	IviRFSigGen»Analog Modulation»AM»AM
AM Source [AM]	IviRFSigGen»Analog Modulation»AM»AM
AM Scaling [AM]	IviRFSigGen»Analog Modulation»AM»AM
AM External Coupling [AM]	IviRFSigGen»Analog Modulation»AM»AM Coupling [AM]
AM Nominal Voltage [AM]	IviRFSigGen»Analog Modulation»AM»AM Voltage [AM]
AM Depth [AM]	IviRFSigGen»Analog Modulation»AM»AM
FM	
FM Enabled [FM]	IviRFSigGen»Analog Modulation»FM»FM
FM Source [FM]	IviRFSigGen»Analog Modulation»FM»FM
FM External Coupling [FM]	IviRFSigGen»Analog Modulation»FM»FM Coupling [FM]

FM Nominal Voltage [FM]

[IviRFSigGen»Analog Modulation»FM»FM Voltage \[FM\]](#)

FM Deviation [FM]

[IviRFSigGen»Analog Modulation»FM»FM](#)

PM

PM Enabled [PM]

[IviRFSigGen»Analog Modulation»PM»PM](#)

PM Source [PM]

[IviRFSigGen»Analog Modulation»PM»PM](#)

PM External Coupling [PM]

[IviRFSigGen»Analog Modulation»PM»PM Coupling \[PM\]](#)

PM Nominal Voltage [PM]

[IviRFSigGen»Analog Modulation»PM»PM Voltage \[PM\]](#)

PM Deviation [PM]

[IviRFSigGen»Analog Modulation»PM»PM](#)

Source

Modulation Source Count [MS]

[IviRFSigGen»Analog Modulation»Source Source Count \[MS\]](#)

Pulse Modulation

Pulse Modulation Enabled [PULM]

[IviRFSigGen»Pulse Modulation»Pulse Modulation Enabled \[PULM\]](#)

Pulse Modulation Source [PULM]

[IviRFSigGen»Pulse Modulation»Pulse Modulation Source \[PULM\]](#)

Pulse Modulation External Polarity [PULM]

[IviRFSigGen»Pulse Modulation»Pulse Modulation External Polarity \[PULM\]](#)

LF Generator

Active LF Generator [LF]

[IviRFSigGen»LF G
LF Generator \[LF\]](#)

LF Generator Count [LF]

[IviRFSigGen»LF G
Generator Count \[L](#)

LF Generator Frequency [LF]

[IviRFSigGen»LF G
Generator Frequen](#)

LF Generator Waveform [LF]

[IviRFSigGen»LF G
Generator Wavefor](#)

LF Generator Outputs

LF Generator Output Amplitude [LFO]

[IviRFSigGen»LF G
Generator Outputs:
Output Amplitude \[I](#)

LF Generator Output Enabled [LFO]

[IviRFSigGen»LF G
Generator Outputs:
Output Enabled \[LF](#)

Pulse Generator

Pulse Internal Trigger Period [PG]

[IviRFSigGen»Pulse
Generator»Pulse Ir
Period \[PG\]](#)

Pulse Width [PG]

[IviRFSigGen»Pulse
Generator»Pulse W](#)

Pulse Gating Enabled [PG]

[IviRFSigGen»Pulse
Generator»Pulse G
\[PG\]](#)

Pulse Trigger Source [PG]

[IviRFSigGen»Pulse
Generator»Pulse T
\[PG\]](#)

Pulse External Trigger Slope [PG]

[IviRFSigGen»Pulse
Generator»Pulse E
Slope \[PG\]](#)

Pulse External Trigger Delay [PG]

[IviRFSigGen»Pulse
Generator»Pulse E
Delay \[PG\]](#)

Double Pulse Generators

Pulse Double Enabled [DPG]

[IviRFSigGen»Pulse Generator»Double Generators»Pulse I \[DPG\]](#)

Pulse Double Delay [DPG]

[IviRFSigGen»Pulse Generator»Double Generators»Pulse I \[DPG\]](#)

Pulse Generator Output

Pulse Output Enabled [PGO]

[IviRFSigGen»Pulse Generator»Pulse G Output»Pulse Outp \[PGO\]](#)

Pulse Output Polarity [PGO]

[IviRFSigGen»Pulse Generator»Pulse G Output»Pulse Outp](#)

Sweep

Sweep Mode [SWP]

[IviRFSigGen»Swee \[SWP\]](#)

Sweep Trigger Source [SWP]

[IviRFSigGen»Swee Source \[SWP\]](#)

Frequency Sweep

Frequency Sweep Start [FSW]

[IviRFSigGen»Swee Sweep»Frequency \[FSW\]](#)

Frequency Sweep Stop [FSW]

[IviRFSigGen»Swee Sweep»Frequency \[FSW\]](#)

Frequency Sweep Time [FSW]

[IviRFSigGen»Swee Sweep»Frequency \[FSW\]](#)

Power Sweep

Power Sweep Start [PSW]

[IviRFSigGen»Swee Sweep»Power Swe](#)

Power Sweep Stop [PSW] [IviRFSigGen»Sweep Sweep»Power Sweep](#)

Power Sweep Time [PSW] [IviRFSigGen»Sweep Sweep»Power Sweep](#)

Frequency Step

Frequency Step Start [FST] [IviRFSigGen»Sweep Step»Frequency Step](#)

Frequency Step Stop [FST] [IviRFSigGen»Sweep Step»Frequency Step](#)

Frequency Step Scaling [FST] [IviRFSigGen»Sweep Step»Frequency Step](#)

Frequency Step Size [FST] [IviRFSigGen»Sweep Step»Frequency Step](#)

Frequency Step Single Step Enabled [FST] [IviRFSigGen»Sweep Step»Frequency Step Enabled \[FST\]](#)

Frequency Step Dwell [FST] [IviRFSigGen»Sweep Step»Frequency Step](#)

Power Step

Power Step Start [PST] [IviRFSigGen»Sweep Step»Power Step S](#)

Power Step Stop [PST] [IviRFSigGen»Sweep Step»Power Step S](#)

Power Step Size [PST] [IviRFSigGen»Sweep Step»Power Step S](#)

Single Step Enabled [PST] [IviRFSigGen»Sweep Step»Single Step E](#)

Power Step Dwell [PST] [IviRFSigGen»Sweep Step»Power Step L](#)

List

List Selected Name [LST] [IviRFSigGen»Sweep Selected Name \[LS](#)

List Single Step Enabled [LST] [IviRFSigGen»Sweep](#)

List Dwell [LST]

[Single Step Enable
IviRFSigGen»Sweet
\[LST\]](#)

ALC

ALC Source [ALC]

[IviRFSigGen»ALC»
\[ALC\]](#)

ALC Bandwidth [ALC]

[IviRFSigGen»ALC»
\[ALC\]](#)

Reference Oscillator

Reference Oscillator Source [RO]

[IviRFSigGen»Refer
Oscillator»Referenc
Source \[RO\]](#)

Reference Oscillator External Frequency [RO]

[IviRFSigGen»Refer
Oscillator»Referenc
External Frequency](#)

IQ

IQ Enabled [MIQ]

[IviRFSigGen»IQ»IC](#)

IQ Nominal Voltage [MIQ]

[IviRFSigGen»IQ»IC
Voltage \[MIQ\]](#)

IQ Source [MIQ]

[IviRFSigGen»IQ»IC](#)

IQSwap Enabled [MIQ]

[IviRFSigGen»IQ»IC
\[MIQ\]](#)

IQ Impairment

IQ Impairment Enabled [IQI]

[IviRFSigGen»IQ»IC
Impairment Enable](#)

IQ I Offset [IQI]

[IviRFSigGen»IQ»IC
I Offset \[IQI\]](#)

IQ Q Offset [IQI]

[IviRFSigGen»IQ»IC
Q Offset \[IQI\]](#)

IQ Ratio [IQI]

[IviRFSigGen»IQ»IC
Ratio \[IQI\]](#)

IQ Skew [IQI]

[IviRFSigGen»IQ»IC](#)

[Skew \[IQI\]](#)

ARB Generator

ARB Clock Frequency [ARB]

[IviRFSigGen»ARB
Clock Frequency \[A](#)

ARB Filter Frequency [ARB]

[IviRFSigGen»ARB
Filter Frequency \[A](#)

Waveform

ARB Selected Waveform [ARB]

[IviRFSigGen»ARB
Generator»Wavefo
Selected Waveform](#)

ARB Max Number Waveforms [ARB]

[IviRFSigGen»ARB
Generator»Wavefo
Number Waveform:](#)

ARB Waveform Quantum [ARB]

[IviRFSigGen»ARB
Generator»Wavefo
Waveform Quantun](#)

ARB Waveform Size Min [ARB]

[IviRFSigGen»ARB
Generator»Wavefo
Waveform Size Mir](#)

ARB Waveform Size Max [ARB]

[IviRFSigGen»ARB
Generator»Wavefo
Waveform Size Ma](#)

Trigger

ARB Trigger Source [ARB]

[IviRFSigGen»ARB
Generator»Trigger»
Source \[ARB\]](#)

ARB External Trigger Slope [ARB]

[IviRFSigGen»ARB
Generator»Trigger»
Trigger Slope \[ARB](#)

Digital Modulation

DigitalModulationBase Standard Count [DMB]

[IviRFSigGen»Digit»
Modulation»Digital
Standard Count \[DI](#)

DigitalModulationBase Selected Standard [DMB]	IviRFSigGen»DigitalModulation»DigitalSelected Standard
DigitalModulationBase Data Source [DMB]	IviRFSigGen»DigitalModulation»DigitalData Source [DMB]
DigitalModulationBase PRBS Type [DMB]	IviRFSigGen»DigitalModulation»DigitalPRBS Type [DMB]
DigitalModulationBase Selected Bit Sequence [DMB]	IviRFSigGen»DigitalModulation»DigitalSelected Bit Sequence
DigitalModulationBase Clock Source [DMB]	IviRFSigGen»DigitalModulation»DigitalClock Source [DME]
DigitalModulationBase External Clock Type [DMB]	IviRFSigGen»DigitalModulation»DigitalExternal Clock Type
CDMA	
CDMA Standard Count [CDMA]	IviRFSigGen»CDMAStandard Count [CDMA]
CDMA Selected Standard [CDMA]	IviRFSigGen»CDMASelected Standard
CDMA Test Model Count [CDMA]	IviRFSigGen»CDMATest Model Count [CDMA]
CDMA Selected Test Model [CDMA]	IviRFSigGen»CDMASelected Test Model
CDMA Clock Source [CDMA]	IviRFSigGen»CDMAClock Source [CDMA]
Trigger	
CDMA Trigger Source [CDMA]	IviRFSigGen»CDMATrigger Source [CDMA]
CDMA External Trigger Slope [CDMA]	IviRFSigGen»CDMAExternal Trigger Slope

TDMA

TDMA Standard Count [TDMA]

[IviRFSigGen»TDM.
Standard Count \[TI](#)

TDMA Selected Standard [TDMA]

[IviRFSigGen»TDM.
Selected Standard](#)

TDMA Frame Count [TDMA]

[IviRFSigGen»TDM.
Count \[TDMA\]](#)

TDMA Selected Frame [TDMA]

[IviRFSigGen»TDM.
Selected Frame \[TI](#)

TDMA Clock Source [TDMA]

[IviRFSigGen»TDM.
Source \[TDMA\]](#)

TDMA Clock Type [TDMA]

[IviRFSigGen»TDM.
Type \[TDMA\]](#)

Trigger

TDMA Trigger Source [TDMA]

[IviRFSigGen»TDM.
Trigger Source \[TD](#)

TDMA External Trigger Slope [TDMA]

[IviRFSigGen»TDM.
External Trigger Slo](#)

IviRFSigGen Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviRFSigGen Errors	0xBFFFA2001 to 0xBFFFA3FFF
IviRFSigGen Warnings	0x3FFFA2001 to 0x3FFFA3FFF
IVI Specific Driver Errors	0xBFFFA4000 to 0xBFFFA5FFF
IVI Specific Driver Warnings	0x3FFFA4000 to 0x3FFFA5FFF
IVI Errors	0xBFFFA0000 to 0xBFFFA1FFF
IVI Warnings	0x3FFFA0000 to 0x3FFFA1FFF
Common Instrument Driver Errors	0xBFFFC0000 to 0xBFFFCFFFF
Common Instrument Driver Warnings	0x3FFFC0000 to 0x3FFFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0xFFFFFFF

The IviRFSigGen class driver defines the error codes shown in the following table in addition to the IVI defined error codes.

IvIRFSigGen Error Codes

Error	Value	Message
IVIRFSIGGEN_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001	The trigger source not software triggered
IVIRFSIGGEN_ERROR_MAX_TIME_EXCEEDED	0xBFFA2003	Maximum time exceeded before operation completed
IVIRFSIGGEN_ERROR_LIST_UNKNOWN	0xBFFA200A	List is unknown

Related Topic

[IVI Status Codes](#)

IviRFSigGen VIs

Expand this book to view an alphabetized list of IviRFSigGen VIs.

IviRFSigGen Calibrate IQ [MIQ]

IviRFSigGenModulateIQ Capability Group

This VI calibrates the IQ modulator.



 **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Clear All Arb Waveforms [ARB]

IviRFSigGenArbGenerator Capability Group

This VI deletes all the currently defined arb waveforms.



 **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Clear All DigitalModulationBase Bit Sequences [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI clears (deletes) all named bit sequences.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Clear All Lists [LST]

IviRFSigGenList Capability Group

This VI deletes all lists from the pool of defined lists. The following VIs create list names: , , or .



 **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

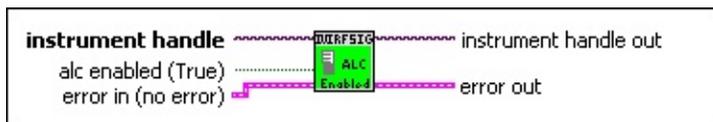
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure ALC Enabled

IviRFSigGenBase Capability Group

This VI enables the Automatic Level Control.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **alc enabled** Specifies whether to enable or disable Automatic Level Control (ALC). The driver uses this value to set the property.

Valid Values:

TRUE (1) - Enables the Automatic Level Control (ALC)

FALSE (0) - Disables the Automatic Level Control (ALC)

Default Value: TRUE (1)

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

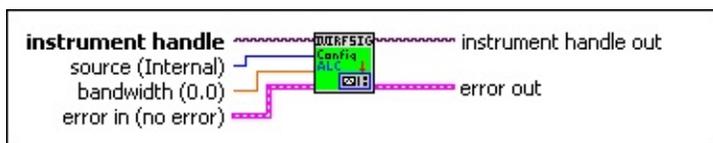
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure ALC [ALC]

IviRFSigGenALC Capability Group

This VI configures the ALC (Automatic Level Control) of the signal generator's RF output.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the controlling voltage for the Automatic Level Control. The RF level at the sensor point is held constant. The driver uses this value to set the IviRFSigGen>>ALC>>ALC Source property.

Defined Values:

IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL

- The ALC is controlled by an internal measurement source.

IVIRFSIGGEN_VAL_ALC_SOURCE_EXTERNAL

- The ALC is controlled by an external voltage.

Default Value: IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL

DBL **bandwidth** Specifies the bandwidth of Automatic Level Control (ALC). Narrow bandwidth improves noise and allows AM with modulation frequencies beyond bandwidth frequency. The value is coerced to reflect the ability of the current RF signal generator. The driver uses this value to set the IviRFSigGen>>ALC>>ALC Bandwidth property.

Units: Hertz

Default Value: 0.0

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

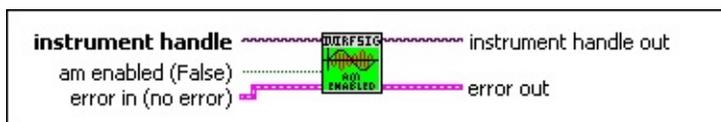
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure AM Enabled [AM]

IviRFSigGenModulateAM Capability Group

This VI configures the signal generator to apply amplitude modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **am enabled** Specifies whether the signal generator applies amplitude modulation to the RF output signal or not. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>AM>>AM Enabled property.

Valid Values:

TRUE (1) - Amplitude modulation On

FALSE (0) - Amplitude modulation Off

Default Value: FALSE (0)

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

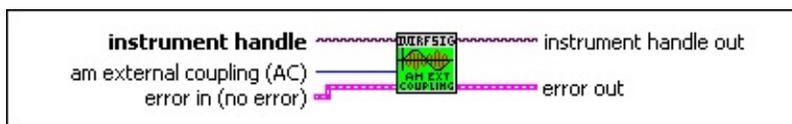
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure AM External Coupling [AM]

IviRFSigGenModulateAM Capability Group

This VI configures the coupling of an external source for amplitude modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **am external coupling** Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>AM>>AM External Coupling property.

Defined Values:

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_AC - AC Coupling

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_DC - DC Coupling

Default Value:

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_AC

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



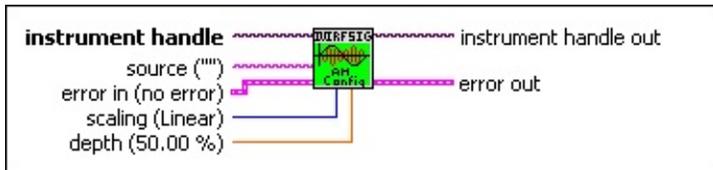
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure AM [AM]

IviRFSigGenModulateAM Capability Group

This VI configures the modulation source, scaling, and depth for the signal generator's amplitude modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **source** Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>AM>>AM Source property.

Default Value: ""

I32 **scaling** Specifies linear or logarithmic attenuation for amplitude modulation. The unit of the IviRFSigGen>>Analog Modulation>>AM>>AM Depth property is changed with this setting. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>AM>>AM Scaling property.

Defined Values:

IVIRFSIGGEN_VAL_AM_SCALING_LINEAR

IVIRFSIGGEN_VAL_AM_SCALING_LOGARITHMIC

Default Value: IVIRFSIGGEN_VAL_AM_SCALING_LINEAR

D6L **depth** Specifies the extent of modulation the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The driver uses this value to set the IviRFSigGen>>Analog

Modulation>>AM>>AM Depth property.

Units: If the IviRFSigGen>>Analog Modulation>>AM>>AM Scaling property is set to Linear, then the units are percent (%). If the IviRFSigGen>>Analog Modulation>>AM>>AM Scaling property is set to logarithmic, then the units are dBm.

Default Value: 0.0



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



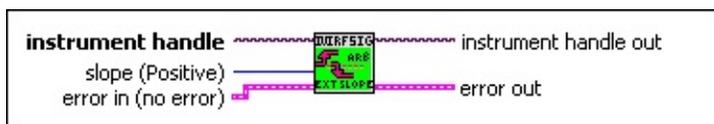
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Arb External Trigger Slope [ARB]

IviRFSigGenArbGenerator Capability Group

This VI configures the trigger event to occur on the rising or falling edge of the input signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **slope** Specifies whether the trigger event occurs on the rising or falling edge of the input signal. The driver uses this value to set the IviRFSigGen>>ARB Generator>>Trigger>>ARB External Trigger Slope property.

Defined Values:

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_POSITIVE
- Enables rising edge triggering.

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_NEGATIVE
- Enables falling edge triggering.

Default Value:

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_POSITIVE

Err **error in (no error)** The error in cluster can accept error information will from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from t

[IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



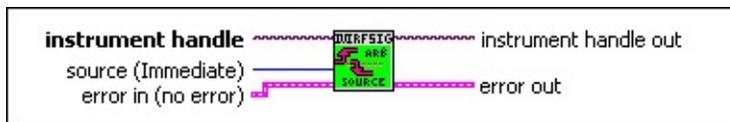
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Arb Trigger Source [ARB]

IviRFSigGenArbGenerator Capability Group

This VI configures the trigger source for the waveform generation. The output waveform is generated continuously if the source is immediate. Otherwise, the output is triggered.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies how the Arb waveform is started. The driver uses this value to set the IviRFSigGen>>ARB Generator>>Trigger>>ARB Trigger Source property.

Defined Values:

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IMMEDIATE -
The
ARB generator system does not wait for a trigger. The
ARB runs continuously.

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_EXTERNAL - The
sweep is started with an external signal.

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SOFTWARE -
The
sweep is started with a software programmable trigger.

Default Value:

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IMMEDIATE

ECH **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



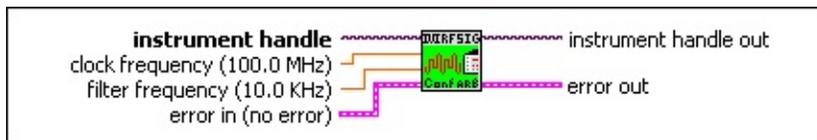
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Arb [ARB]

IviRFSigGenArbGenerator Capability Group

This VI configures the ARB generator by specifying the sample frequency and filter frequency.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **clock frequency** Specifies the sample frequency. The waveform is generated with this clock frequency. The driver uses this value to set the IviRFSigGen>>ARB Generator>>ARB Clock Frequency property.

Units: Hertz

Default Value: 100.0 MHz

DBL **filter frequency** Specify the filter frequency which is the cut-off frequency of the low pass filter used for anti aliasing the output waveform. The filter frequency normally is lower than the clock frequency.

The driver uses this value to set the IviRFSigGen>>ARB Generator>>ARB Filter Frequency property.

Units: Hertz

Default Value: 10.0 KHz

ERR **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



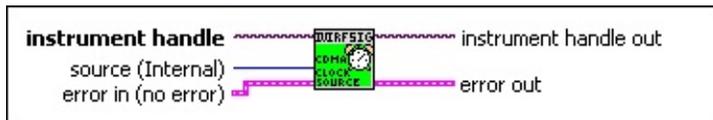
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure CDMA Clock Source [CDMA]

IviRFSigGenCDMABase Capability Group

This VI configures the CDMA clock source.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the clock signal used to generate the digital modulation according to the selected standard. The driver uses this value to set the IviRFSigGen>>CDMA>>CDMA Clock Source property.

Defined Values:

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INTERNAL
- The internal clock generator is used.

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_EXTERNAL
- A connected external clock generator (bit or symbol clock frequency) is used.

Default Value:

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INTERNAL

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure CDMA External Trigger Slope [CDMA]

IviRFSigGenCDMABase Capability Group

This VI configures the trigger event to occur on the rising or falling edge of the input signal. This setting is used only if IviRFSigGen>>CDMA>>Trigger>>CDMA Trigger Source is set to External.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **slope** Specifies whether the trigger event occurs on the rising or falling edge of the input signal. The driver uses this value to set the IviRFSigGen>>CDMA>>Trigger>>CDMA External Trigger Slope prop

Defined Values:

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_POSIT
- Enables rising edge triggering.

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_NEGA
- Enables falling edge triggering.

Default Value:

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_POSIT

E31 **error in (no error)** The error in cluster can accept error information will from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O

instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



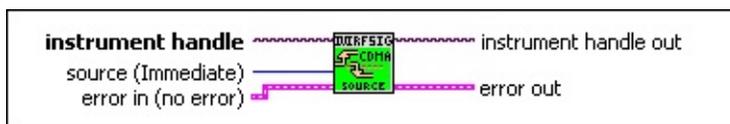
error out The error out cluster passes error or warning information out of the VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure CDMA Trigger Source [CDMA]

IviRFSigGenCDMABase Capability Group

This VI configures the CDMA trigger source for starting or synchronizing the generation of the channel codings.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the trigger signal that starts the channel coding generation. The driver uses this value to set the IviRFSigGen>>CDMA>>Trigger>>CDMA Trigger Source property.

Defined Values:

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_IMMEDIATE -
The
CDMA generator system does not wait for a trigger.
Each channel coding is run continuously.

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_EXTERNAL -
Each
channel coding is started with an external signal.

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SOFTWARE -
Each
channel coding is started with a software programmable
trigger.

Default Value:

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_IMMEDIATE

[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



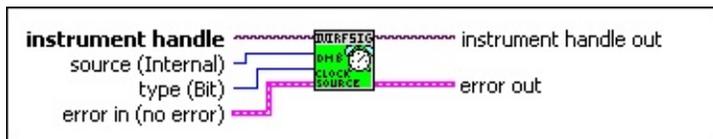
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure DigitalModulationBase Clock Source [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI configures the DigitalModulationBase clock source.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the clock signal used to generate the according to the selected standard. The driver uses this value to set IviRFSigGen>>Digital Modulation>>DigitalModulationBase Clock Source

Defined Values:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE
- The internal clock generator is used.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE
- A connected external clock generator (bit or symbol clock frequency)

Default Value:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE

I32 **type** Specifies the type of the external clock signal used to generate This value is used only if the IviRFSigGen>>Digital Modulation>>Digital ModulationBase Clock Source property is set to External. The driver uses this value to IviRFSigGen>>Digital Modulation>>DigitalModulationBase External (

Defined Values:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_C
- The external clock frequency is equal to the bit clock frequency of the

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_C
- The external clock frequency is equal to the symbol clock frequency of the modulation.

Default Value:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_C

 [error in \(no error\)](#) The error in cluster can accept error information will be called. Use this information to decide if any functionality should be bypassed by errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information displayed.

 **instrument handle out** The instrument handle that you obtain from the [Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

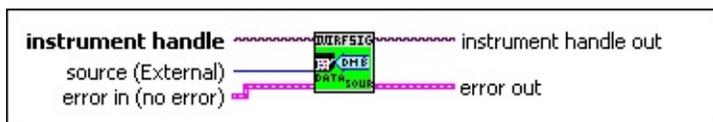
 [error out](#) The error out cluster passes error or warning information out to other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information displayed.

IviRFSigGen Configure DigitalModulationBase Data Source [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI configures the source of the data for the digital modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of data. The data is used to modulate the signal according to the standard selected with the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Selected Standard property. The value to set the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Selected Standard property.

Defined Values:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_EXTERNAL
- The data from an external device connected to the instrument is used as the data source.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_INTERNAL
- The internal PRBS (Pseudo Random Binary Sequence) generator is used as the data source.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_CONSTANT
- A constant bit sequence is used as data source and repeated continuously.

Default Value:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_EXTERNAL

Err **error in (no error)** The error in cluster can accept error information with the error in (no error) property.

previously called. Use this information to decide if any functionality should be disabled in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out to other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure DigitalModulationBase PRBS Type [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI configures the type of the PRBS used as data for the digital modulation. The setting is used only if IviRFSigGen>>Digital Modulation>>DigitalModulationBase Data Source is set to PRBS.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **type** Specifies the type of the PRBS as defined in the CCITT-V.52 standard. The PRBS (Pseudo Random Binary Sequence) is used only if the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Data Source property is set to PRBS. The driver uses this value to set the IviRFSigGen>>Digital Modulation>>DigitalModulationBase PRBS Type property.

Defined Values:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_9
- Length of PRBS sequence is $2^{E9}-1$.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_11
- Length of PRBS sequence is $2^{E11}-1$.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_15
- Length of PRBS sequence is $2^{E15}-1$.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_19

- Length of PRBS sequence is $2^{E16}-1$.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_
- Length of PRBS sequence is $2^{E20}-1$

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_
- Length of PRBS sequence is $2^{E21}-1$.

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_
- Length of PRBS sequence is $2^{E23}-1$.

Default Value:

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_

 [error in \(no error\)](#) The error in cluster can accept error information with VIs previously called. Use this information to decide if any functionality can be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

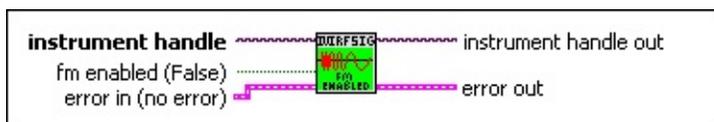
 [error out](#) The error out cluster passes error or warning information out to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure FM Enabled [FM]

IviRFSigGenModulateFM Capability Group

This VI configures the signal generator to apply frequency modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **fm enabled** Specifies whether the signal generator applies frequency modulation to the RF output signal or not. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>FM>>FM Enabled property.

Valid Values:

TRUE (1) - Frequency modulation On
FALSE (0) - Frequency modulation Off

Default Value: FALSE (0)

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

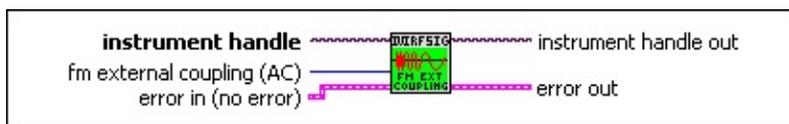
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure FM External Coupling [FM]

IviRFSigGenModulateFM Capability Group

This VI configures the coupling of an external source for frequency modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **fm external coupling** Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>FM>>FM External Coupling property.

Defined Values:

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_AC - AC Coupling

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_DC - DC Coupling

Default Value:

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_AC

Err **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



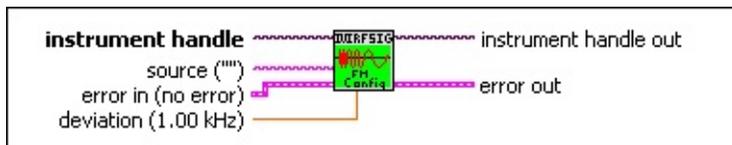
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure FM [FM]

IviRFSigGenModulateFM Capability Group

This VI configures the modulation deviation and modulation source for the signal generator's frequency modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **source** Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>FM>>FM Source property.

Default Value: ""

DBL **deviation** Specifies the extent of modulation (peak frequency deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>FM>>FM Deviation property.

Units: Hertz

Default Value: 0.0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



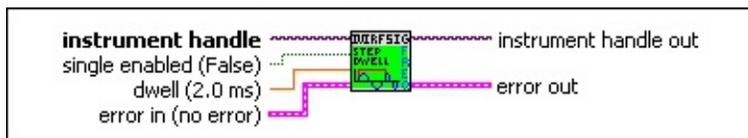
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Frequency Step Dwell [FST]

IviRFSigGenFrequencyStep Capability Group

This VI configures how the frequency sweep advances.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **single enabled** Specify if the frequency sweep advances after the next trigger event (TRUE) or immediately after the dwell time ends (FALSE). The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_SINGLE_STEP_ENABLED property.

Valid Values: TRUE; FALSE

Default Value: FALSE

DBL **dwell** Specify the duration time of one step. Dwell time starts immediate after a trigger or the next step; no settling time is added. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_DWELL property.

Units: Seconds

Default Value: 2.0 ms

Note:

This property is ignored if the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_SINGLE_STEP_ENABLED property is

set to TRUE.

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

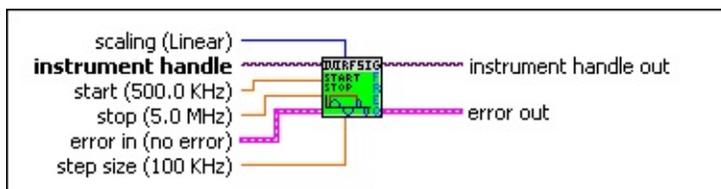
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Frequency Step Start Stop [FST]

IviRFSigGenFrequencyStep Capability Group

This VI configures the settings that control the step frequencies of the generator's RF output signal. These settings are start and stop frequency, step size, and lin/log scaling. If the stop frequency is less than the start frequency, the frequency decreases during the sweep.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **start** Specifies the start frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_STAR property.

Units: Hertz

Default Value: 500.0 KHz

DBL **stop** Specifies the stop frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_STOP property.

Units: Hertz

Default Value: 5.0 MHz

I32 **scaling** Specifies the spacing of the steps. The driver uses this value

to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_SCALING property.

Defined Values:

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR - Linear scaling

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMI - Logarithmic scaling

Default Value:

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR

 **step size** Specifies the step size. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_SIZE property.

Units:

The units are Hz when the IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_SCALING property is set to IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR

The value is unitless (factor) when the

IviRFSigGen>>Sweep>>Frequency

Step>>Frequency_STEP_SCALING property is set to

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMI

Default Value: 100 KHz

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from t [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **[error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

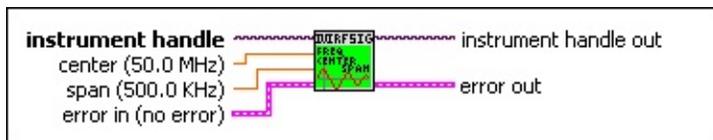
IviRFSigGen Configure Frequency Sweep Center Span [FSW]

IviRFSigGenFrequencySweep Capability Group

This VI configures the center frequency and span for a frequency sweep. This VI modifies the start and stop properties as follows:

IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_START =
Center - Span / 2

IviRFSigGen>>Sweep>>Frequency Step>>Frequency_STEP_STOP =
Center + Span / 2



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **center** Specifies the center frequency for a frequency sweep.

Units: Hertz

Default Value: 50.0 MHz

DBL **span** Specifies the frequency span for a frequency sweep.

Units: Hertz

Default Value: 500.0 KHz

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Frequency Sweep Start Stop [FSW]

IviRFSigGenFrequencySweep Capability Group

This VI configures the start and stop frequencies for a frequency sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **start** Specifies the start frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Sweep>>Frequency_SWEEP_START property.

Units: Hertz

Default Value: 100.0 KHz

DBL **stop** Specifies the stop frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Sweep>>Frequency_SWEEP_STOP property.

Units: Hertz

Default Value: 100.0 MHz

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Frequency Sweep Time [FSW]

IviRFSigGenFrequencySweep Capability Group

This VI configures the duration of one frequency sweep.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **time** Specifies the duration of one sweep from start to stop frequency. The driver uses this value to set the IviRFSigGen>>Sweep>>Frequency Sweep>>Frequency_SWEEP_TIME property.

Units: Seconds

Default Value: 1.0 s

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

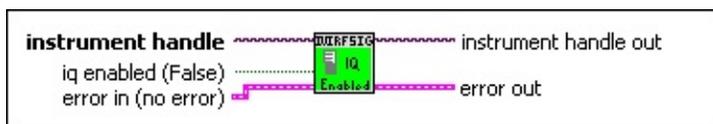
The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

IviRFSigGen Configure IQ Enabled [MIQ]

IviRFSigGenModulateIQ Capability Group

This VI configures the signal generator to apply IQ (vector) modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **iq enabled** Enables IQ (vector) modulation of the output RF signal. The driver uses this value to set the IviRFSigGen>>IQ>>IQ Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

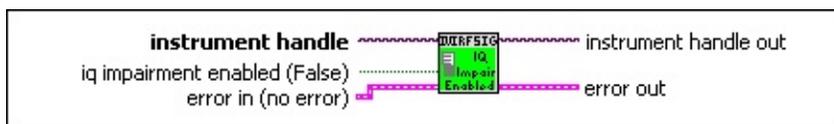
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure IQ Impairment Enabled [IQI]

IviRFSigGenIQImpairment Capability Group

This VI enables the IQ (vector) modulation to allow controlled impairment for test or external corrections.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **iq impairment enabled** Enables IQ impairment. The driver uses this value to set the IviRFSigGen>>IQ>>IQ Impairment>>IQ Impairment Enabled property.

Valid Values: TRUE; FALSE

When set to TRUE, the following impairment properties are applied:

IviRFSigGen>>IQ>>IQ Impairment>>IQ I Offset
IviRFSigGen>>IQ>>IQ Impairment>>IQ Q Offset
IviRFSigGen>>IQ>>IQ Impairment>>IQ Ratio
IviRFSigGen>>IQ>>IQ Impairment>>IQ Skew

Default Value: FALSE

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O

instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



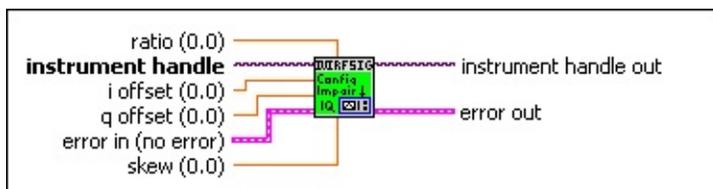
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure IQ Impairment [IQI]

IviRFSigGenIQImpairment Capability Group

This VI configures the settings that simulate or correct impairment for the signal generator's IQ modulation. These settings are only used if the IviRFSigGen>>IQ>>IQ Impairment>>IQ Impairment Enabled property is set to TRUE.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.
- DBL** **i offset** Specifies an origin offset voltage to the I signal. The range of allowable values is -100% to +100%. The driver uses this value to set the IviRFSigGen>>IQ>>IQ Impairment>>IQ I Offset property.

Units: Percent (%)

Default Value: 0.0

- DBL** **q offset** Specifies an origin offset voltage to the Q signal. The range of allowable values is -100% to +100%. The driver uses this value to set the IQ_Q_OFFSET property.

Units: Percent (%)

Default Value: 0.0

- DBL** **ratio** Specifies the gain imbalance between the I and Q channels. For no imbalance this value is set to 0 %. The driver uses this value to set the IviRFSigGen>>IQ>>IQ Impairment>>IQ Ratio property.

Units: Percent (%)

Default Value: 0.0



skew Specifies the adjustment of the phase angle between the I and Q vectors. If this skew is zero, the phase angle is 90 degrees. The driver uses this value to set the `IviRFSigGen>>IQ>>IQ Impairment>>IQ Skew` property.

Units: Degrees

Default Value: 0.0



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



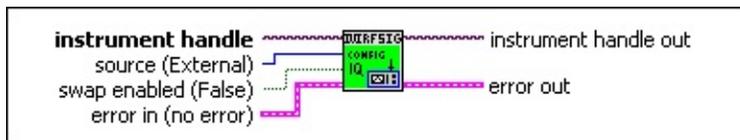
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure IQ [MIQ]

IviRFSigGenModulateIQ Capability Group

This VI configures the signal generator to apply IQ (vector) modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the signal that the signal generator uses for IQ modulation. The driver uses this value to set the IviRFSigGen>>IQ>>IQ Source property.

Defined Values:

IVIRFSIGGEN_VAL_IQ_SOURCE_DIGITAL_MODULATION_BASE
- The signal generator uses the internally generated digital modulation signal to apply IQ modulation to the output RF signal.

IVIRFSIGGEN_VAL_IQ_SOURCE_CDMA_BASE - The signal generator uses the internally generated CDMA signal to apply IQ modulation to the output RF signal.

IVIRFSIGGEN_VAL_IQ_SOURCE_TDMA_BASE - The signal generator uses the internally generated TDMA signal to apply IQ modulation to the output RF signal.

IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL - The signal generator uses data from an external source for IQ modulation.

IVIRFSIGGEN_VAL_IQ_SOURCE_ARB_GENERATOR - The signal generator uses the internally generated Arb signal to apply IQ modulation to the output RF signal.

Default Value: IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL

 **swap enabled** Enables the inverse phase rotation of the IQ signal by swapping the I and Q inputs. The driver uses this value to set the IviRFSigGen>>IQ>>IQSwap Enabled property.

Valid Values:

TRUE - The RF signal generator applies non-inverse phase rotation of the IQ signal.

FALSE - The RF signal generator applies inverse phase rotation of the IQ signal.

Default Value: FALSE

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

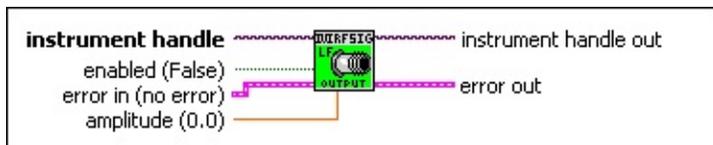
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure LF Generator Output [LFO]

IviRFSigGenLFGeneratorOutput Capability Group

This VI sets the output voltage and enables the LF generator within the RF signal generator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **amplitude** Specifies the output voltage the of the LF generator. The driver uses this value to set the IviRFSigGen>>LF Generator>>LF Generator Outputs>>LF Generator Output Amplitude property.

Units: Volts peak to peak.

Default Value: 0.0

TF **enabled** Specifies whether the LF generator applies an output signal or not. The driver uses this value to set the IviRFSigGen>>LF Generator>>LF Generator Outputs>>LF Generator Output Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



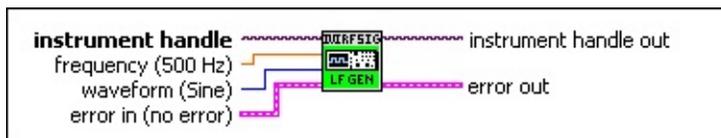
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure LF Generator [LF]

IviRFSigGenLFGenerator Capability Group

This VI configures the LF generators output frequency and waveform.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **frequency** Specifies the frequency of the active LF generator. The driver uses this value to set the IviRFSigGen>>LF Generator>>LF Generator Frequency property.

Units: Hertz

Default Value: 500 Hz

I32 **waveform** Specifies the waveform of the active LF generator. The driver uses this value to set the IviRFSigGen>>LF Generator>>LF Generator Waveform property.

Defined Values:

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SINE
- Sinusoid waveform.

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SQUARE
- Square waveform.

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_TRIANGLE
- Triangle waveform.

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_UP

- Rising ramp waveform.

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_DOWN

- Falling ramp waveform.

Default Value:

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SINE

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

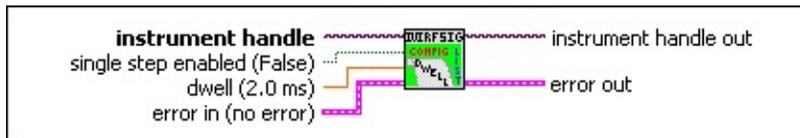
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure List Dwell [LST]

IviRFSigGenList Capability Group

This VI configures how list stepping advances.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.
- TF** **single step enabled** Specifies if the list step advances after the next trigger event (TRUE) or immediately after the dwell time ends (FALSE).

Valid Values:

TRUE - The list advances when the next trigger event occurs.
FALSE - The list will advance immediately after the dwell time ends.

The driver uses this value to set the IviRFSigGen>>Sweep>>List>>List Single Step Enabled property.

Default Value: FALSE

- DBL** **dwell** Specifies the duration time of one step. The driver uses this value to set the IviRFSigGen>>Sweep>>List>>List Dwell property.

Units: Seconds

Default Value: 2.0 ms

 **Notes**

(1) This property is ignored if List Single Step Enabled is set to TRUE.

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

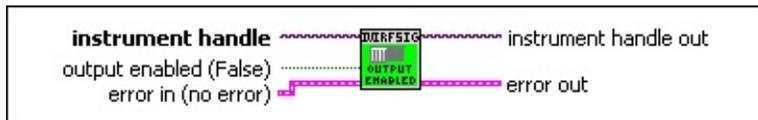
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Output Enabled

IviRFSigGenBase Capability Group

This VI enables the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **output enabled** Specifies whether to enable or disable the RF output signal. The driver uses this value to set the IviRFSigGen>>RF>>Ouptut Enabled property.

Valid Values:

TRUE (1) - Output enabled
FALSE (0) - Output disabled

Default Value: FALSE (0)

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

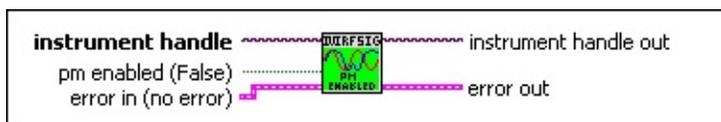
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure PM Enabled [PM]

IviRFSigGenModulatePM Capability Group

This VI configures the signal generator to apply phase modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **pm enabled** Specifies whether the signal generator applies phase modulation to the RF output signal or not. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>PM>>PM Enabled property.

Valid Values:

TRUE (1) - Phase modulation On

FALSE (0) - Phase modulation Off

Default Value: FALSE (0)

E/W **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

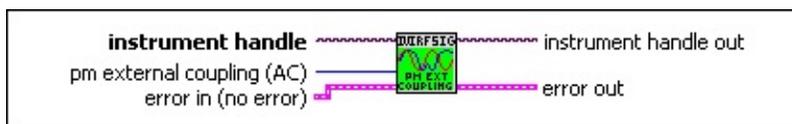
E/W **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure PM External Coupling [PM]

IviRFSigGenModulatePM Capability Group

This VI configures the coupling of an external source for phase modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **pm external coupling** Specifies the coupling of the external source of the modulating signal. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>PM>>PM External Coupling property.

Defined Values:

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_AC
- The external source is coupled for AC only.
IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_DC
- The external source is coupled for both DC and AC.

Default Value:

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_AC

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

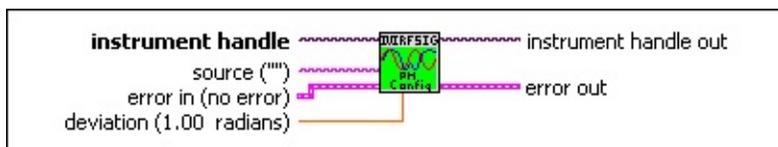
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure PM [PM]

IviRFSigGenModulatePM Capability Group

Configures the property that control the signal generator's phase modulation. The properties are the modulation deviation and the modulating source(s).

This VI configures the modulation deviation and modulation source for the signal generator's phase modulation.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **source** Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>PM>>PM Source property.

Default Value: ""

DBL **deviation** Specifies the extent of modulation (peak phase deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The driver uses this value to set the IviRFSigGen>>Analog Modulation>>PM>>PM Deviation property.

Units: radians

Default Value: 0.0

ERR **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



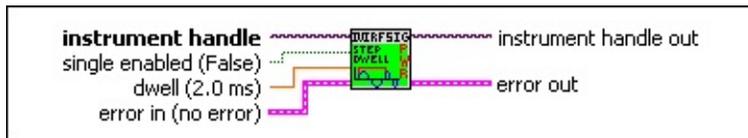
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Power Step Dwell [PST]

IviRFSigGenPowerStep Capability Group

This VI configures how the power sweep advances.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **single enabled** Specifies if the frequency sweep advances after the next trigger event (TRUE) or immediately after the dwell time ends (FALSE). The driver uses this value to set the IviRFSigGen>>Sweep>>Power Step>>Single Step Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

DBL **dwell** Specifies the duration time of one step. Dwell time starts immediate after a trigger or the next step; no settling time is added. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Step>>Power Step Dwell property.

Units: Seconds

Default Value: 2.0 ms

Note:

This property is ignored if the IviRFSigGen>>Sweep>>Power Step>>Single Step Enabled property is set to TRUE.

Err [error in \(no error\)](#) The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



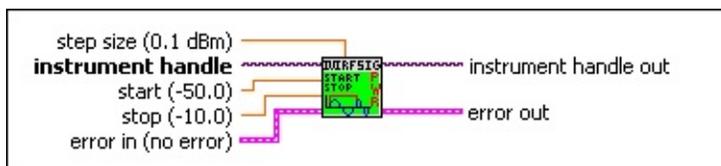
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Power Step Start Stop [PST]

IviRFSigGenPowerStep Capability Group

This VI configures the settings that control the power steps of the generator's RF output signal. These settings are start and stop power and step size. If the stop power is less than the start power, the power decreases in value during the sweep



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **start** Specifies the start power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Step>>Power Step Start property.

Units: dBm

Default Value: -50.0

DBL **stop** Specifies the start power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Step>>Power Step Stop property.

Units: dBm

Default Value: -10.0

DBL **step size** Specifies the step size. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Step>>Power Step Size property.

Units: dBm

Default Value: 0.1 dBm



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



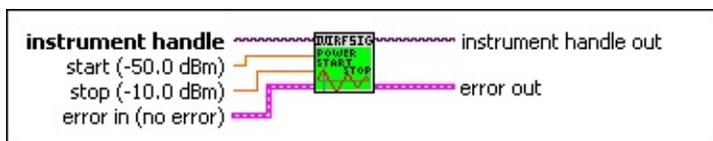
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Power Sweep Start Stop [PSW]

IviRFSigGenPowerSweep Capability Group

This VI configures the start and stop power for a power sweep. If the stop power is less than the start power, the power decreases in value during the sweep.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **start** Specifies the start power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Sweep>>Power Sweep Start property.

Units: dBm

Default Value: -50.0 dBm

DBL **stop** Specifies the stop power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Sweep>>Power Sweep Stop property.

Units: dBm

Default Value: -10.0 dBm

ERR **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



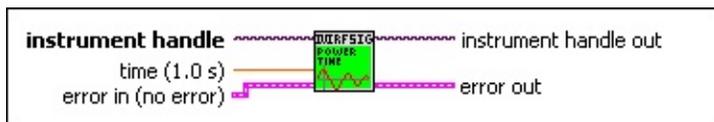
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Power Sweep Time [PSW]

IviRFSigGenPowerSweep Capability Group

This VI configures the duration of one power sweep.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **time** Specifies the duration of one sweep from start to stop power. The driver uses this value to set the IviRFSigGen>>Sweep>>Power Sweep>>Power Sweep Time property.

Units: Seconds

Default Value: 1.0 s

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

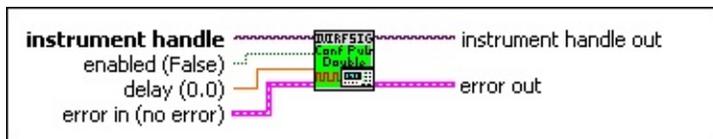
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse Double [DPG]

IviRFSigGenPulseDoubleGenerator Capability Group

This VI sets the double pulse state and delay for the pulse generator within the RF signal generator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **enabled** Enables double pulse mode. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Double Pulse Generators>>Pulse Double Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

DBL **delay** Specifies the delay of the second pulse. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Double Pulse Generators>>Pulse Double Delay property.

Units: Seconds

Default Value: 0.0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



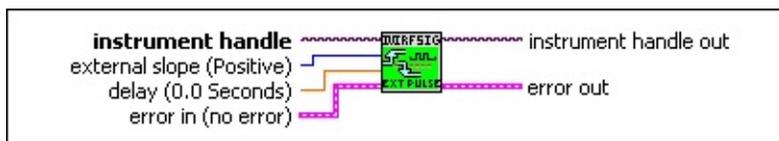
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse External Trigger [PG]

IviRFSigGenPulseGenerator Capability Group

This VI configures the triggering of the pulse generator within the RF signal generator. This VI specifies the external trigger slope and the delay time for starting the pulse after the trigger pulse.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **external slope** Specifies whether the event occurs on the rising or falling edge of the input signal. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse External Trigger Slope property.

Defined Values:

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_POSITIVE
- Enables rising edge triggering.

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_NEGATIVE
- Enables falling edge triggering.

Default Value:

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_POSITIVE

DBL **delay** Specifies the delay for starting the output pulse with respect to trigger input.

Units: Seconds

Default Value: 0.0

 [error in \(no error\)](#) The error in cluster can accept error information from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs. The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

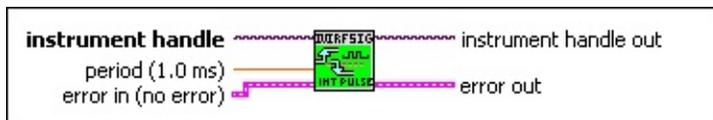
 [error out](#) The error out cluster passes error or warning information out of the VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse Internal Trigger [PG]

IviRFSigGenPulseGenerator Capability Group

This VI specifies the time period (repetition rate) of the pulse generator when using the internal trigger (free run) mode.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **period** Specifies the period of the pulse generator output signal when the property IviRFSigGen>>Pulse Generator>>Pulse Trigger Source is set to Internal. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Internal Trigger Period property.

Units: Seconds

Default Value: 1.0 ms

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

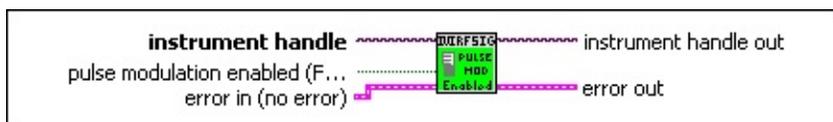
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse Modulation Enabled [PULM]

IviRFSigGenModulatePulse Capability Group

This VI configures the signal generator to apply pulse modulation to the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **pulse modulation enabled** Specifies whether the signal generator applies pulse modulation to the RF output signal or not. The driver uses this value to set the IviRFSigGen>>Pulse Modulation>>Pulse Mod. Enabled property.

Valid Values:

TRUE (1) - Enables pulse modulation

FALSE (0) - Disables pulse modulation

Default Value: FALSE

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

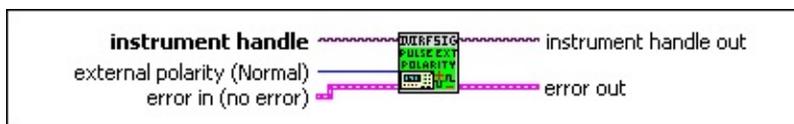
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse Modulation External Polarity [PULM]

IviRFSigGenModulatePulse Capability Group

This function specifies the polarity of the external source signal.



I/O **instrument handle** The instrument handle that you obtain from the [Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **external polarity** Specifies the polarity of the external source signal. uses this value to set the IviRFSigGen>>Pulse Modulation>>Pulse M Polarity property.

Defined Values:

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARIT
- The signal generator modulates the carrier signal with normal pulse polarity. Increasing the positive pulse voltage level results in higher RF level.

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARIT
- The signal generator modulates the carrier signal with inverted pulse polarity. Increasing the positive pulse voltage level results in lower RF level.

Default Value:

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARIT

Err **error in (no error)** The error in cluster can accept error information wii previously called. Use this information to decide if any functionality sh bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more info about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

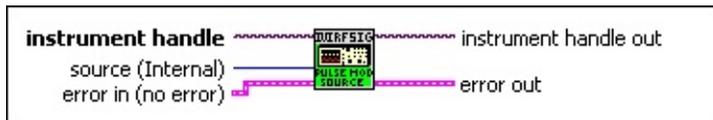
 **error out** The error out cluster passes error or warning information out to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more info about the error displayed.

IviRFSigGen Configure Pulse Modulation Source [PULM]

IviRFSigGenModulatePulse Capability Group

This VI sets the source for pulse modulation of the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the signal that is used as the modulating signal. The driver uses this value to set the IviRFSigGen>>Pulse Modulation>>Pulse Mod. Source property.

Defined Values:

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_INTERNAL
- The internal pulse generator is used for modulation.

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_EXTERNAL
- An external generator is used for modulation

Default Value:

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_INTERNAL

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error out** The error out cluster passes error or warning information

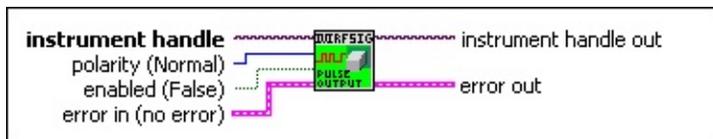
out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse Output [PGO]

IviRFSigGenPulseGeneratorOutput Capability Group

This VI configures the output and polarity of the pulse generator within the RF signal generator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **polarity** Specifies the polarity of the output signal. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Generator Output>>Pulse Output Polarity property.

Defined Values:

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_NORMAL
- Normal polarity

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_INVERSE
- Inverted polarity

Default Value:

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_NORMAL

TFI **enabled** Enables the pulse generator external output. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Generator Output>>Pulse Output Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



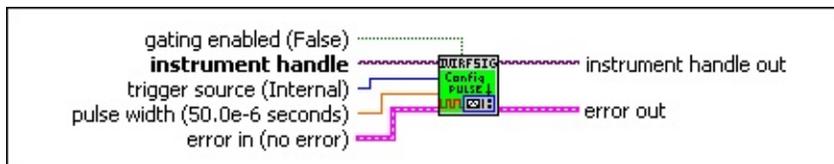
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Pulse [PG]

IviRFSigGenPulseGenerator Capability Group

This VI configures the trigger source, pulse width, and gating enabled for the pulse generator within the RF signal generator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **trigger source** Specifies the source of the signal the pulse generator uses to generate one pulse. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Trigger Source property.

Defined Values:

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_INTERNAL
- No external trigger is used. The pulse period is specified by the IviRFSigGen>>Pulse Generator>>Pulse Internal Trigger Period property.

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_EXTERNAL
- The pulse is started with a trigger after the delay time specified by the IviRFSigGen>>Pulse Generator>>Pulse External Trigger Delay property.

Default Value:

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_INTERNAL
DBL **pulse width** Specifies the width of the output pulse. The driver

uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Width property.

Units: Seconds

Default Value: 50.0e-6 seconds

 **gating enabled** Enables pulse gating. The driver uses this value to set the IviRFSigGen>>Pulse Generator>>Pulse Gating Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

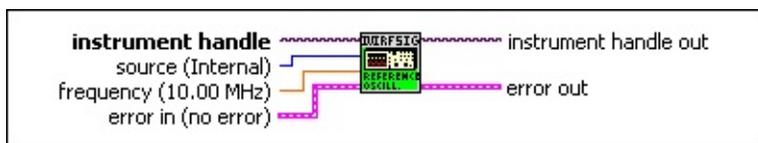
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Reference Oscillator [RO]

IviRFSigGenReferenceOscillator Capability Group

This VI configures the signal generator's reference oscillator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the reference frequency source used to generate the exact RF output frequency. The driver uses this value to set the IviRFSigGen>>Reference Oscillator>>Reference Oscillator Source property.

Defined Values:

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_INTEF
- The internal reference oscillator is used.

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTE
- An external reference oscillator is used.

Default Value:

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_INTEF

DBL **frequency** Specifies the frequency of the external signal, which is used as a reference for internal RF frequency generation. The driver uses this value to set the IviRFSigGen>>Reference Oscillator>>Reference Oscillator External Frequency property.

Units: Hertz

Default Value: 10.0E6

Note:

This value is used only if the IviRFSigGen>>Reference Oscillator>>Reference Oscillator Source property is set to External.

 [error in \(no error\)](#) The error in cluster can accept error information will from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

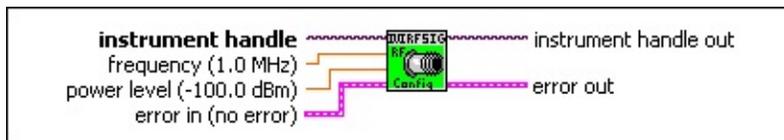
 [error out](#) The error out cluster passes error or warning information out VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure RF

IviRFSigGenBase Capability Group

This VI configures the frequency and the power level of the RF output signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

DBL **frequency** Specifies the frequency of the generated RF signal. The driver uses this value to set the property.

Units: Hertz

Default Value: 1.0 MHz

DBL **power level** Specifies the power level of the generated RF signal. The driver uses this value to set the property.

Units: dBm

Default Value: 0.0

ERR **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

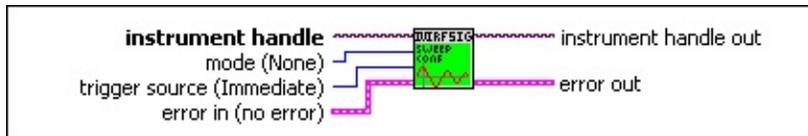
ERR **[error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure Sweep

IviRFSigGenSweep Capability Group

This VI configures the signal generator sweep mode and trigger source.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **mode** Specifies the sweep mode applied to the output signal. The driver uses this value to set the IviRFSigGen>>Sweep>>Sweep Mode property.

Defined Values:

IVIRFSIGGEN_VAL_SWEEP_MODE_NONE - The RF output of the signal generator is a non-swept signal (Continuous Wave) using the frequency and power/level specified in the .

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_SWEEP - The signal generator sweeps the RF output signal's frequency in an analog form (non-stepped).

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_SWEEP - The signal generator sweeps the RF output signal's power in an analog form (non-stepped).

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_STEP - The signal generator sweeps the RF output signals frequency in steps.

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_STEP - The signal generator sweeps the RF output signals power level in steps.

IVIRFSIGGEN_VAL_SWEEP_MODE_LIST - The signal generator uses two lists with frequency and power level values to sweep the RF output signals.

Default Value: IVIRFSIGGEN_VAL_SWEEP_MODE_NONE

 **trigger source** Specifies the trigger used to start a sweep operation. The driver uses this value to set the IviRFSigGen>>Sweep>>Sweep Trigger Source property.

Defined Values:

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_IMMEDIATE -
The sweep system does not wait for a trigger of any kind, so it is running continuously.

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL -
The sweep is started with an external signal.

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SOFTWARE -
The sweep is started with a software programmable trigger.

Default Value:

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_IMMEDIATE
 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With](#)

[Options](#) VIs.



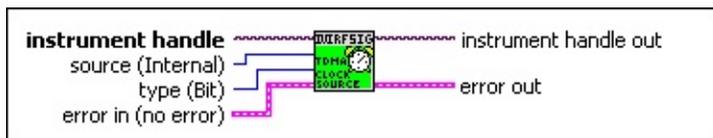
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure TDMA Clock Source [TDMA]

IviRFSigGenTDMABase Capability Group

This VI configures the TDMA clock source.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the clock signal used to generate the digital modulation according to the selected standard. The driver uses this value to set the IviRFSigGen>>TDMA>>TDMA Clock Source property.

Defined Values:

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INTERNAL
- The internal clock generator is used.

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_EXTERNAL
- A connected external clock generator (bit or symbol clock frequency) is used.

Default Value:

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INTERNAL

I32 **type** Specifies the type of the external clock signal used to generate the digital modulation. This value is used only if the IviRFSigGen>>TDMA>>TDMA Clock Source property is set to External.

Type is ignored if Source is set to Internal.

Defined Values:

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_BIT
- The external clock frequency is equal to the bit clock frequency of the digital modulation.

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SYMBOL
- The external clock frequency is equal to the symbol clock frequency of the digital modulation.

Default Value:

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_BIT

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

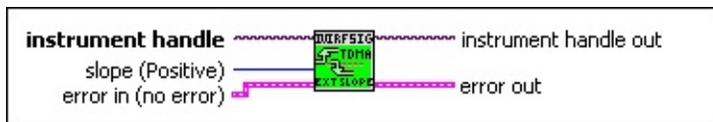
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure TDMA External Trigger Slope [TDMA]

IviRFSigGenTDMABase Capability Group

This VI configures the trigger event to occur on the rising or falling edge of the input signal. This setting is used only if IviRFSigGen>>TDMA>>Trigger>>TDMA Trigger Source is set to External



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **slope** Specifies whether the trigger event occurs on the rising or falling edge of the input signal. The driver uses this value to set the IviRFSigGen>>TDMA>>Trigger>>TDMA External Trigger Slope prop

Defined Values:

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_POSIT
- Enables rising edge triggering.

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_NEGA
- Enables falling edge triggering.

Default Value:

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_POSIT

Err **error in (no error)** The error in cluster can accept error information wii from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from t

[IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



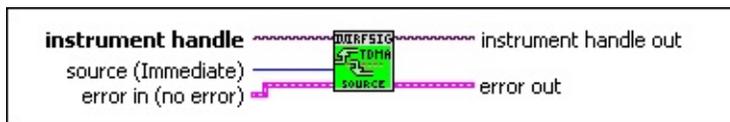
[error out](#) The error out cluster passes error or warning information out a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Configure TDMA Trigger Source [TDMA]

IviRFSigGenTDMABase Capability Group

This VI configures the TDMA trigger source for starting or synchronizing the generation of the frames/slots.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **source** Specifies the source of the trigger signal that starts the frame/slots generation. The driver uses this value to set the IviRFSigGen>>TDMA>>Trigger>>TDMA Trigger Source property.

Defined Values:

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_IMMEDIATE -
The
TDMA generator system does not wait for a trigger.
Each frame is run continuously.

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_EXTERNAL -
Each
frame is started with an external signal.

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SOFTWARE -
Each
frame is started with a software programmable
trigger.

Default Value:

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_IMMEDIATE

[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



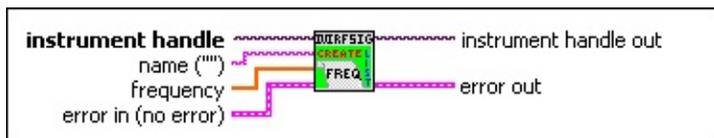
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Create Frequency List [LST]

IviRFSigGenList Capability Group

This VI creates a named list of frequency values.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** The name of the list to be created.

Default Value: ""

I32 **length** The number of values in the list array.

Default Value: 0

abc **frequency** Pass the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the Length parameter.

Units: Hertz

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

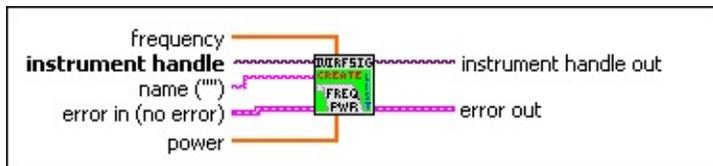
I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Create Frequency Power List [LST]

This VI creates a named list of frequency and power values.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** The name of the list to be created.

Default Value: ""

I32 **length** The number of values in the list array.

Default Value: 0

abc **frequency** Pass the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the Length parameter.

Units: Hertz

abc **power** Pass the array of power values to become elements of the list. The array must have at least as many elements as the value in the Length parameter.

Units: dBm

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Create Power List [LST]

IviRFSigGenList Capability Group

This VI creates a named list of power values.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** The name of the list to be created.

Default Value: ""

I32 **length** The number of values in the list array.

Default Value: 0

abc **power** Pass the array of power values to become elements of the list. The array must have at least as many elements as the value in the Length parameter.

Units: dBm

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Disable All Modulation

IviRFSigGenBase Capability Group

This VI disables all currently enabled modulations (e.g. analog, pulse, IQ, and digital modulation).



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E/W **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E/W **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

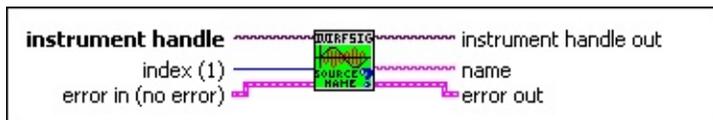
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get Analog Modulation Source Name [MS]

IviRFSigGenAnalogModulationSource Capability Group

This VI returns the specific driver defined analog modulation source name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>Analog Modulation>>Source>>Modulation Source Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one analog modulation source, i.e. the IviRFSigGen>>Analog Modulation>>Source>>Modulation Source Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** The index of the analog modulation source whose name is returned. The value must be greater than or equal to one and less than or equal to IviRFSigGen>>Analog Modulation>>Source>>Modulation Source Count.

Note:

Zero is not a legal value.

Default Value: 1

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0



name The analog modulation source name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get CDMA Standard Name [CDMA]

IviRFSigGenCDMABase Capability Group

This VI returns the specific driver defined CDMA standard name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>CDMA>>CDMA Standard Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one CDMA standard, i.e. the IviRFSigGen>>CDMA>>CDMA Standard Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** Specifies the index of the CDMA Standard Name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [lviRFSigGen Initialize](#) or [lviRFSigGen Initialize With Options](#) VIs.



name The CDMA standard name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

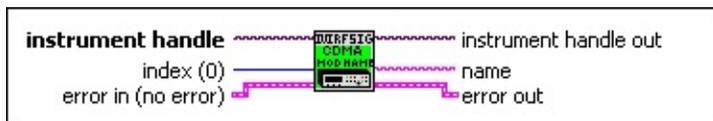
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get CDMA Test Model Name [CDMA]

IviRFSigGenCDMABase Capability Group

This VI returns the specific driver defined CDMA test model name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>CDMA>>CDMA Test Model Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one CDMA test model, i.e. the IviRFSigGen>>CDMA>>CDMA Test Model Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** Specifies the index of the CDMA test model name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

E32 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [lviRFSigGen Initialize](#) or [lviRFSigGen Initialize With Options](#) VIs.



name The CDMA test model name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get Digital Modulation Base Standard Name [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI returns the specific driver defined DigitalModulationBase standard name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Standard Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one DigitalModulationBase standard, i.e. the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Standard Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** Specifies the index of the DigitalModulationBase Name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

E-T **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **name** The DigitalModulationBase name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get LF Generator Name [LF]

IviRFSigGenLFGenerator Capability Group

This VI returns the specific driver defined LF generator source name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>LF Generator>>LF Generator Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one LF generator source, i.e. the IviRFSigGen>>LF Generator>>LF Generator Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** The index of the analog modulation source whose name is returned. The value must be greater than or equal to one and less than or equal to IviRFSigGen>>LF Generator>>LF Generator Count.

Note:

Zero is not a legal value.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

Err **error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **name** The LF generator source name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.

Default Value: ""

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

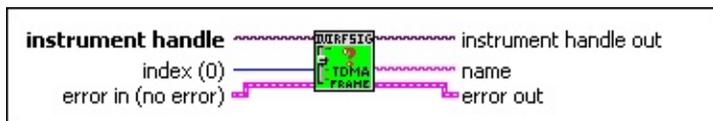
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get TDMA Frame Name [TDMA]

IviRFSigGenTDMABase Capability Group

This VI returns the specific driver defined TDMA frame name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviRFSigGen>>TDMA>>TDMA Frame Count property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one TDMA frame, i.e. the IviRFSigGen>>TDMA>>TDMA Frame Count property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** Specifies the index of the TDMA frame name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [lviRFSigGen Initialize](#) or [lviRFSigGen Initialize With Options](#) VIs.



name The TDMA frame name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Get TDMA Standard Name [TDMA]

IviRFSigGenTDMABase Capability Group

This VI returns the specific driver defined TDMA standard name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the TDMA_STANDARD_COUNT property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one CDMA standard, i.e. the TDMA_STANDARD_COUNT property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **index** Specifies the index of the TDMA Standard Name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [lviRFSigGen Initialize](#) or [lviRFSigGen Initialize With Options](#) VIs.



name The TDMA standard name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



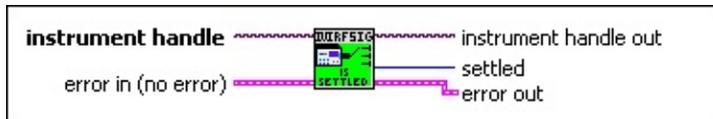
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Is Settled

IviRFSigGenBase Capability Group

This VI queries if the RF output signal is currently settled.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

TF **settled** Returns TRUE if the output signal is in settled, FALSE otherwise.

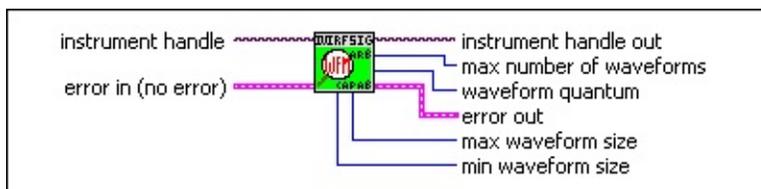
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Query Arb Waveform Capabilities [ARB]

IviRFSigGenArbGenerator Capability Group

This VI returns the arb generator settings that are related to creating arbitrary waveforms. These properties are the maximum number of waveforms, waveform quantum, minimum waveform size, and maximum waveform size.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **max number of waveforms** Returns the maximum number of waveforms the instrument can hold in the memory. The number may be dependent on the length of the waveform already in the pool of waveforms saved in the instrument. The driver returns the value of the IviRFSigGen>>ARB Generator>>Waveform>>ARB Max Number Waveforms property.

I32 **waveform quantum** Returns the waveform quantum where the waveform length is a multiple of this quantum. If the waveform quantum is 1, there is no restriction on the waveform length other

than min and max size. The driver returns value of the IviRFSigGen>>ARB Generator>>Waveform>>ARB Waveform Quantum property.

 **min waveform size** Returns the minimum size of the waveform length in number of samples. The waveform length must be equal or greater than this min size. If the minimum size is 1, there is no restriction on the waveform length other than max size and quantum. The driver returns the value of the IviRFSigGen>>ARB Generator>>Waveform>>ARB Waveform Size Min property.

 **max waveform size** Returns the maximum waveform length in the number of samples. The waveform length must be equal or less than this maximum size. The driver returns the value of the IviRFSigGen>>ARB Generator>>Waveform>>ARB Waveform Size Max property.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Reset Frequency Step [FST]

IviRFSigGenFrequencyStep Capability Group

This VI resets the current frequency step to the frequency step start value.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Reset List [LST]

IviRFSigGenList Capability Group

This VI resets the current list to the first entry value.



 **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Reset Power Step [PST]

IviRFSigGenPowerStep Capability Group

This VI resets the current power step to the power step start value.



 **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

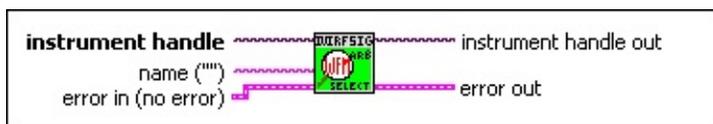
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select Arb Waveform [ARB]

IviRFSigGenArbGenerator Capability Group

This VI sets a named waveform to be the active waveform. Create arb waveform names using the .



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specifies the selected waveform from the pool of available waveforms. The driver uses this value to set the IviRFSigGen>>ARB Generator>>Waveform>>ARB Selected Waveform property.

Default Value: ""

E+ **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E+ **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select CDMA Standard [CDMA]

IviRFSigGenCDMABase Capability Group

This VI specifies the standard used by the instrument. To obtain a list of available standards, call the `IviRFSigGenInitialize`. Many instrument settings are affected by selecting an instrument standard. These settings include: modulation type, bit clock frequency and filter together with the associated filter parameters.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specify the standard to be used by the instrument. The driver uses this value to set the `IviRFSigGen>>CDMA>>CDMA Selected Standard` property.

Default Value: ""

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

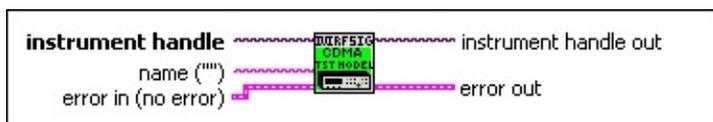
err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select CDMA Test Model [CDMA]

IviRFSigGenCDMABase Capability Group

This VI specifies the channel coding used for the digital modulation. To obtain a list of available CDMA test models for channel coding, call the .



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specifies the actual test model used by the instrument. The driver uses this value to set the IviRFSigGen>>CDMA>>CDMA Selected Test Model property.

Default Value: ""

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

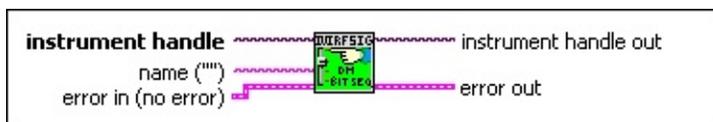
err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select DigitalModulationBase Bit Sequence [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI sets a bit sequence to be used as digital modulation data. Create bit sequences using the . The bit sequence is set only if the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Data Source property is set to BitSequence.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specifies name of the bit sequence (stream) used as data for digital modulation. The sequence is used only if the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Data Source property is set to Bit Sequence. The driver uses this value to set the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Selected Bit Sequence property.

Default Value: ""

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



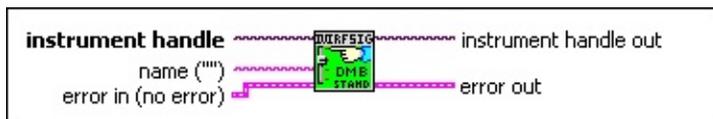
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select DigitalModulationBase Standard [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI specifies the actual standard used by the instrument. To obtain a list of available standards, call the . Many instrument settings are affected by selecting an instrument standard. These settings include: coding, mapping, symbol rate or bit clock frequency, filter together with the associated filter parameters, and FSK deviation or ASK depth (in the case of FSK or ASK modulation).



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specifies the standard to be used by the instrument. The driver uses this value to set the IviRFSigGen>>Digital Modulation>>DigitalModulationBase Selected Standard property.

Default Value: ""

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select List [LST]

IviRFSigGenList Capability Group

This VI sets a named list to be the active list. Create list names using the following VIs: , , or .



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specify the name of the list to make this list active. The name must be one of the lists created. The driver uses this value to set the IviRFSigGen>>Sweep>>List>>List Selected Name property.

Default Value: ""

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

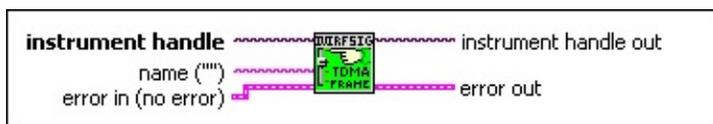
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select TDMA Frame [TDMA]

IviRFSigGenTDMABase Capability Group

This VI specifies the frame used for framed digital modulation. To obtain a list of available TDMA frames, call the .



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specifies the actual frames/slot used by the instrument. It is selected with one of the names from TDMA Frame Names property. The frames/slot are generated only, if TDMA Framing Enabled is set to TRUE. This property is ignored if TDMA Framing Enabled is set to FALSE. The driver uses this value to set the IviRFSigGen>>TDMA>>TDMA Selected Frame property.

Default Value: ""

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

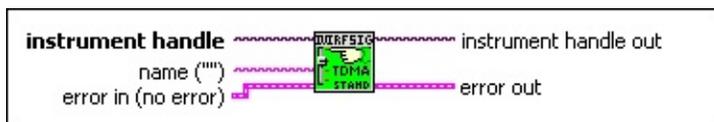
err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Select TDMA Standard [TDMA]

IviRFSigGenTDMABase Capability Group

This VI specifies the standard used by the instrument. To obtain a list of available standards, call the . Many instrument settings are affected by selecting an instrument standard. These settings include: coding, mapping, symbol rate or bit clock frequency, filter together with the associated filter parameters, and FSK deviation or ASK depth (in case of FSK or ASK modulation).



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Specify the standard to be used by the instrument. The driver uses this value to set the IviRFSigGen>>TDMA>>TDMA Selected Standard property.

Default Value: ""

E+! **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

E+! **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

IviRFSigGen Send Software Trigger [SWT]

This VI sends a command to trigger the RF signal generator. Call this VI if you set a trigger source to a software trigger value. Below is a list of properties and VIs which can set the trigger source to a software trigger.

VIs:

Properties:

IviRFSigGen>>ARB Generator>>Trigger>>ARB Trigger Source

IviRFSigGen>>Sweep>>Sweep Trigger Source

IviRFSigGen>>CDMA>>Trigger>>CDMA Trigger Source

IviRFSigGen>>TDMA>>Trigger>>TDMA Trigger Source



Notes

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

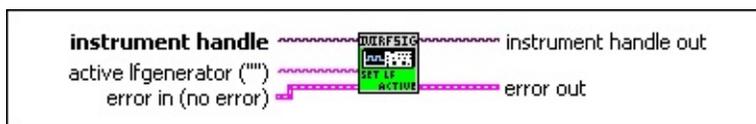
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Set Active LF Generator [LF]

IviRFSigGenLFGenerator Capability Group

This VI selects one of the available LF generator sources, and makes it the active LFGenerator.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **active lfgenerator** Specifies the LF generator which is currently active. The values for this attribute correspond to the LFGenerator repeated capability. The driver uses this value to set the IviRFSigGen>>LF Generator>>Active LF Generator property.

Default Value: ""

E+H **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

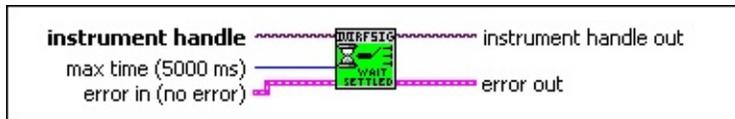
E+H **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Wait Until Settled

IviRFSigGenBase Capability Group

This VI waits until the state of the RF output signal has settled.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

I32 **max time** Specifies the maximum time to wait for the output to settle.

If the output does not settle within this time interval, the VI returns the IVIRFSIGGEN_ERROR_MAX_TIME_EXCEEDED (0xBFFA2003) error code.

Units: milliseconds

Defined Values:

IVIRFSIGGEN_VAL_MAX_TIME_INFINITE (-1)
IVIRFSIGGEN_VAL_MAX_TIME_IMMEDIATE (0)

Default Value: 5000 (ms)

Notes

(1) The Maximum Time parameter applies only to this VI. It has no effect on other timeout parameters or properties.

Err [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



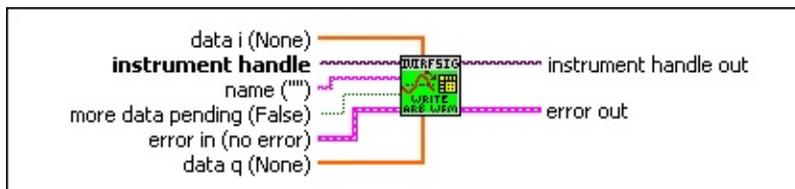
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Write Arb Waveform [ARB]

IviRFSigGenArbGenerator Capability Group

This VI stores the transmitted waveform in the drivers's or instrument's memory.



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Pass the name the waveform to write.

Default Value: ""

I32 **number of samples** Pass the number of samples in both I and Q array.

Default Value: 0

abc **data i** Pass the array containing the I part of the waveform.

You must normalize the data points in the array to be between -1.00 and +1.00 of the full I range.

Default Value: None

abc **data q** Pass the array containing the Q part of the waveform.

You must normalize the data points in the array to be between -1.00 and +1.00 of the full Q range.

Default Value: None



more data pending Set the value to TRUE to write more data. To complete the waveform, use the same Name for the next data. Set the value to FALSE when there is no more data to write.

Valid Values:

FALSE - No more data

TRUE - More data

Default Value: No more data (FALSE)



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



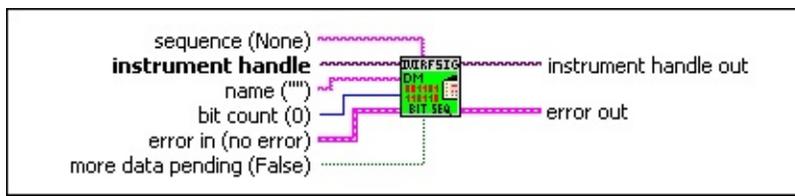
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Write DigitalModulationBase Bit Sequence [DMB]

IviRFSigGenDigitalModulationBase Capability Group

This VI creates a bit sequence for use as the digital modulation data. The sequence is repeated continuously. The sequence string consists of binary values (8 bit in 1 char/byte).



I/O **instrument handle** The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.

abc **name** Pass the name the bit sequence to write.

Default Value: ""

I32 **bit count** Pass the number of bits in the sequence. One char/byte has 8 bits, but the stream of bits does not end necessarily on byte boundaries.

Default Value: 0

abc **sequence** Pass the array the driver will use as a stream of bits. The size of the array is Bit Count.

Default Value: None

TF **more data pending** Set the value to TRUE to write more data. To complete the sequence, use the same Name for the next data. Set the value to FALSE when there is no more data to write.

Valid Values:

FALSE - No more data

TRUE - More data

Default Value: No more data (FALSE)



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviRFSigGen Initialize](#) or [IviRFSigGen Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviRFSigGen Properties

Expand this book to view an alphabetized list of IviRFSigGen properties.

IviRFSigGen Active LF Generator [LF]

IviRFSigGenLFGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Set Active LF Generator [LF]

Description

Specifies the LF generator which is currently active. The values for this property correspond to the LFGenerator repeated capability.

Property Node Path

IviRFSigGen»LF Generator>>Active LF Generator [LF]

IviRFSigGen ALC Bandwidth [ALC]

IviRFSigGenALC Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------

 DBL	R/W	N/A	Up	IviRFSigGen Configure ALC [ALC]
---------------------------------------------------------------------------------------	-----	-----	----	-------------------------------------------------

Description

Specifies the bandwidth of Automatic Level Control.

Property Node Path

IviRFSigGen»ALC>>ALC Bandwidth [ALC]

IviRFSigGen ALC Enabled

IviRFSigGenBase Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------

 R/W	N/A	None	<u>IviRFSigGen Configure ALC Enabled</u>
---------------------------------------------------------------------------------------	-----	------	----------------------------------------------------------

Description

Specifies whether to enable or disable Automatic Level Control (ALC).

Property Node Path

IviRFSigGen»RF>>ALC Enabled

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen ALC Source [ALC]

IviRFSigGenALC Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------

	R/W	N/A	None	IviRFSigGen Configure ALC [ALC]
-----------------------------------------------------------------------------------	-----	-----	------	-------------------------------------------------

Description

Specifies the source of the controlling voltage for the Automatic Level Control.

Property Node Path

IviRFSigGen»ALC>>ALC Source [ALC]

Defined Values

IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_ALC_SOURCE_EXTERNAL

IviRFSigGen AM Depth [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure AM [AM]

Description

Specifies the extend of modulation the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external.

If the [IviRFSigGen>>Analog Modulation>>AM>>AM Scaling \[AM\]](#) property is set to Linear, then the units are percent (%). If the [IviRFSigGen>>Analog Modulation>>AM>>AM Scaling \[AM\]](#) property is set to logarithmic, then the units are dBm.

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM Depth [AM]

IviRFSigGen AM Enabled [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure AM Enabled [AM]

Description

Specifies whether the signal generator applies amplitude modulation to the RF output signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM Enabled [AM]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen AM External Coupling [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure AM External Coupling [AM]

Description

Specifies the coupling of the external source of the modulating signal.

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM External Coupling [AM]

Defined Values

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_AC

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_DC

IviRFSigGen AM Nominal Voltage [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	None

Description

This is the voltage at which the instrument achieves the amount of modulation specified by the [IviRFSigGen>>Analog Modulation>>AM>>AM Depth \[AM\]](#) property.

Note: This property is read only.

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM Nominal Voltage [AM]

IviRFSigGen AM Scaling [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure AM [AM]

Description

Specifies linear or logarithmic attenuation for amplitude modulation. The unit of the [IviRFSigGen>>Analog Modulation>>AM>>AM Depth \[AM\]](#) property is changed with this setting.

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM Scaling [AM]

Defined Values

IVIRFSIGGEN_VAL_AM_SCALING_LINEAR

IVIRFSIGGEN_VAL_AM_SCALING_LOGARITHMIC

IviRFSigGen AM Source [AM]

IviRFSigGenModulateAM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure AM [AM]

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas.

Property Node Path

IviRFSigGen»Analog Modulation>>AM>>AM Source [AM]

IviRFSigGen ARB Clock Frequency [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Arb [ARB]

Description

Specifies the sample frequency. The waveform is generated with this clock frequency.

Property Node Path

IviRFSigGen»ARB Generator>>ARB Clock Frequency [ARB]

IviRFSigGen ARB External Trigger Slope [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Arb External Trigger Slope [ARB]

Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

Property Node Path

IviRFSigGen»ARB Generator>>Trigger>>ARB External Trigger Slope
[ARB]

Defined Values

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_POSITIVE

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_NEGATIVE

IviRFSigGen ARB Filter Frequency [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------



R/W	N/A	UP	IviRFSigGen Configure Arb [ARB]
-----	-----	----	-------------------------------------------------

Description

Specifies the cut-off frequency of the low pass filter. The waveform is filtered before output with this filter for antialiasing. The filter frequency normally is lower than the clock frequency. The units are Hertz.

Property Node Path

IviRFSigGen»ARB Generator>>ARB Filter Frequency [ARB]

IviRFSigGen ARB Max Number Waveforms [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	IviRFSigGen Query Arb Waveform Capabilities [ARB]

Description

Returns the max number of waveforms the instrument can hold in the memory. The number may depend on the length of the waveform already in the pool of waveforms saved in the instrument.

Property Node Path

IviRFSigGen»ARB Generator>>Waveform>>ARB Max Number
Waveforms [ARB]

IviRFSigGen ARB Selected Waveform [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	<u>IviRFSigGen Select Arb Waveform [ARB]</u>

Description

Specifies the selected waveform from the pool of available waveforms.

Property Node Path

IviRFSigGen»ARB Generator>>Waveform>>ARB Selected Waveform
[ARB]

IviRFSigGen ARB Trigger Source [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Arb Trigger Source [ARB]

Description

Specifies how the Arb waveform is started (triggered).

Property Node Path

IviRFSigGen»ARB Generator>>Trigger>>ARB Trigger Source [ARB]

Defined Values

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IMMEDIATE

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_EXTERNAL

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SOFTWARE

IviRFSigGen ARB Waveform Quantum [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	IviRFSigGen Query Arb Waveform Capabilities [ARB]

Description

Returns the waveform quantum where the waveform length is a multiple of this quantum. If the waveform quantum is 1, there is no restriction on the waveform length other than min and max size.

Property Node Path

IviRFSigGen»ARB Generator>>Waveform>>ARB Waveform Quantum
[ARB]

IviRFSigGen ARB Waveform Size Max [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	IviRFSigGen Query Arb Waveform Capabilities [ARB]

Description

Returns the maximum waveform length in the number of samples. The waveform length must be equal or less than this maximum size.

Property Node Path

IviRFSigGen»ARB Generator>>Waveform>>ARB Waveform Size Max
[ARB]

IviRFSigGen ARB Waveform Size Min [ARB]

IviRFSigGenArbGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Query Arb Waveform Capabilities [ARB]

Description

Returns the minimum size of the waveform length in number of samples. The waveform length must equal or greater than this minimum size. If the minimum size is 1, there is no restriction on the waveform length other than max size and quantum.

Property Node Path

IviRFSigGen»ARB Generator>>Waveform>>ARB Waveform Size Min
[ARB]

IviRFSigGen CDMA Clock Source [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure CDMA Clock Source [CDMA]

Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

Property Node Path

IviRFSigGen»CDMA>>CDMA Clock Source [CDMA]

Defined Values

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_EXTERNAL

IviRFSigGen CDMA External Trigger Slope [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure CDMA External Trigger Slope [CDMA]

Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

Property Node Path

IviRFSigGen»CDMA>>Trigger>>CDMA External Trigger Slope [CDMA]

Defined Values

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE

IviRFSigGen CDMA Selected Standard [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select CDMA Standard [CDMA]

Description

Specifies the actual standard used by the instrument. The modulation type, bit clock frequency and filter together with the associated filter settings are set as defined in the selected standard.

Property Node Path

IviRFSigGen»CDMA>>CDMA Selected Standard [CDMA]

IviRFSigGen CDMA Selected Test Model [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select CDMA Test Model [CDMA]

Description

Specifies the actual CDMA test model used by the instrument.

Property Node Path

IviRFSigGen»CDMA>>CDMA Selected Test Model [CDMA]

IviRFSigGen CDMA Standard Count [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Returns the number of CDMA standards available for a particular instrument.

Property Node Path

IviRFSigGen»CDMA>>CDMA Standard Count [CDMA]

IviRFSigGen CDMA Test Model Count [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
I32I	RO	N/A	None	None

Description

Returns the number of CDMA test models available for a particular instrument.

Property Node Path

IviRFSigGen»CDMA>>CDMA Test Model Count [CDMA]

IviRFSigGen CDMA Trigger Source [CDMA]

IviRFSigGenCDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure CDMA Trigger Source [CDMA]

Description

Specifies the source of the trigger signal that starts the channel coding generation.

Property Node Path

IviRFSigGen»CDMA>>Trigger>>CDMA Trigger Source [CDMA]

Defined Values

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_IMMEDIATE

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_EXTERNAL

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SOFTWARE

IviRFSigGen DigitalModulationBase Clock Source [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure DigitalModulationBase Clock Source [DMB]

Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase Clock Source
[DMB]

Defined Values

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_I

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_I

IviRFSigGen DigitalModulationBase Data Source [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure DigitalModulationBase Data Source [DMB]

Description

Specifies the source of data. The data is used to modulate the RF signal according to the standard selected with the DigitalModulationBase Selected Standard property.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase Data Source
[DMB]

Defined Values

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_EX

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_PR

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_BI

IviRFSigGen DigitalModulationBase External Clock Type [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure DigitalModulationBase Clock Source [DMB]

Description

Specifies the type of the external clock signal used to generate the digital modulation. This value is used only if the DigitalModulationBase Clock Source property is set to External.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase External Clock
Type [DMB]

Defined Values

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOC

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOC

IviRFSigGen DigitalModulationBase PRBS Type [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure DigitalModulationBase PRBS Type [DMB]

Description

Specifies the type of the PRBS as defined in the CCITT-V.52 standard. The PRBS (Pseudo Random Binary Sequence) is used only if DigitalModulationBase Data Source is set to PRBS.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase PRBS Type
[DMB]

Defined Values

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS9

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS2

IviRFSigGen DigitalModulationBase Selected Bit Sequence [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select DigitalModulationBase Bit Sequence [DMB]

Description

Specifies name of the bit sequence (stream) used as data for digital modulation. The sequence is used only if DigitalModulationBase Data Source is set to BitSequence.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase Selected Bit Sequence [DMB]

IviRFSigGen DigitalModulationBase Selected Standard [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select DigitalModulationBase Standard [DMB]

Description

Specifies the actual standard used by the instrument. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter. parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase Selected
Standard [DMB]

IviRFSigGen DigitalModulationBase Standard Count [DMB]

IviRFSigGenDigitalModulationBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Specifies the number of DigitalModulationBase standards available for a particular instrument.

Property Node Path

IviRFSigGen»Digital Modulation>>DigitalModulationBase Standard
Count [DMB]

IviRFSigGen FM Deviation [FM]

IviRFSigGenModulateFM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure FM [FM]

Description

Specifies the extent of modulation (peak frequency deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The units are Hz.

Property Node Path

IviRFSigGen»Analog Modulation>>FM>>FM Deviation [FM]

IviRFSigGen FM Enabled [FM]

IviRFSigGenModulateFM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	<u>IviRFSigGen Configure FM Enabled [FM]</u>

Description

Specifies whether the signal generator applies frequency modulation to the RF output signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»Analog Modulation>>FM>>FM Enabled [FM]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen FM External Coupling [FM]

IviRFSigGenModulateFM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure FM External Coupling [FM]

Description

Specifies the coupling of the external source of the modulating signal.

Property Node Path

IviRFSigGen»Analog Modulation>>FM>>FM External Coupling [FM]

Defined Values

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_AC

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_DC

IviRFSigGen FM Nominal Voltage [FM]

IviRFSigGenModulateFM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	None

Description

This is the voltage at which the instrument achieves the amount of modulation specified by the [IviRFSigGen>>Analog Modulation>>FM>>FM Deviation \[FM\]](#) property.

Note: This property is read only.

Property Node Path

IviRFSigGen»Analog Modulation>>FM>>FM Nominal Voltage [FM]

IviRFSigGen FM Source [FM]

IviRFSigGenModulateFM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure FM [FM]

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas.

Property Node Path

IviRFSigGen»Analog Modulation>>FM>>FM Source [FM]

IviRFSigGen Frequency

IviRFSigGenBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure RF

Description

Specifies the frequency of the generated RF output signal.

Property Node Path

IviRFSigGen»RF>>Frequency

IviRFSigGen Frequency Step Dwell [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Step Dwell [FST]

Description

Specifies the duration time of one step. The units are in seconds. Dwell time starts immediate after trigger or next step; no settling time is added.

This property is ignored if Frequency Step Single Step Enabled is set to VI_TRUE.

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Dwell [FST]

IviRFSigGen Frequency Step Scaling [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Frequency Step Start Stop [FST]

Description

Specifies the spacing of the steps.

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Scaling [FST]

Defined Values

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC

IviRFSigGen Frequency Step Single Step Enabled [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Frequency Step Dwell [FST]

Description

Specifies whether the trigger initiates the next step (VI_TRUE), or the next step is taken after dwell time (VI_FALSE).

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Single Step
Enabled [FST]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Frequency Step Size [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Step Start Stop [FST]

Description

Specifies the step size. The units are in Hz when the [IviRFSigGen>>Sweep>>Frequency Step>>Frequency Step Scaling \[FST\]](#) property is set to `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR`. The value is unitless (factor) when the [IviRFSigGen>>Sweep>>Frequency Step>>Frequency Step Scaling \[FST\]](#) property is set to `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC`.

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Size [FST]

IviRFSigGen Frequency Step Start [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Step Start Stop [FST]

Description

Specifies the start frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are in Hz.

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Start [FST]

IviRFSigGen Frequency Step Stop [FST]

IviRFSigGenFrequencyStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Step Start Stop [FST]

Description

Specifies the stop frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are in Hz.

Property Node Path

IviRFSigGen»Sweep>>Frequency Step>>Frequency Step Stop [FST]

IviRFSigGen Frequency Sweep Start [FSW]

IviRFSigGenFrequencySweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Sweep Start Stop [FSW] IviRFSigGen Configure Frequency Sweep Center Span [FSW]

Description

Specifies the start frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are in Hz.

Property Node Path

IviRFSigGen»Sweep>>Frequency Sweep>>Frequency Sweep Start
[FSW]

IviRFSigGen Frequency Sweep Stop [FSW]

IviRFSigGenFrequencySweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Frequency Sweep Start Stop [FSW] IviRFSigGen Configure Frequency Sweep Center Span [FSW]

Description

Specifies the stop frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are in Hz.

Property Node Path

IviRFSigGen»Sweep>>Frequency Sweep>>Frequency Sweep Stop
[FSW]

IviRFSigGen Frequency Sweep Time [FSW]

IviRFSigGenFrequencySweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	<u>IviRFSigGen Configure Frequency Sweep Time [FSW]</u>

Description

Specifies the duration of one sweep from start to stop frequency. The units are in seconds.

Property Node Path

IviRFSigGen»Sweep>>Frequency Sweep>>Frequency Sweep Time
[FSW]

IviRFSigGen IQ Enabled [MIQ]

IviRFSigGenModulateIQ Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure IQ Enabled [MIQ]

Description

Specifies whether the signal generator applies IQ (vector) modulation to the output RF signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»IQ>>IQ Enabled [MIQ]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen IQ Impairment Enabled [IQI]

IviRFSigGenIQImpairment Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure IQ Impairment Enabled [IQI]

Description

Enables IQ impairment. When set to VI_TRUE, the following impairment properties are applied.

IviRFSigGen>>IQ>>IQ Impairment>>IQ I-Offset [IQI]

IviRFSigGen>>IQ>>IQ Impairment>>IQ Q-Offset [IQI]

IviRFSigGen>>IQ>>IQ Impairment>>IQ Ratio [IQI]
IviRFSigGen>>IQ>>IQ Impairment>>IQ Skew [IQI]

Property Node Path

IviRFSigGen»IQ>>IQ Impairment>>IQ Impairment Enabled [IQI]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen IQ Nominal Voltage [MIQ]

IviRFSigGenModulateIQ Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	None

Description

This is the voltage at which the instrument achieves full modulation. The value is calculated by $\text{SQRT}(I^2+Q^2)$.

Property Node Path

IviRFSigGen»IQ>>IQ Nominal Voltage [MIQ]

IviRFSigGen IQ I Offset [IQI]

IviRFSigGenIQImpairment Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure IQ Impairment [IQI]

Description

Specifies an origin offset voltage to the I signal. The range of values allowed is -100% to +100%. The value is expressed as percentage (%).

Property Node Path

IviRFSigGen»IQ>>IQ Impairment>>IQ I Offset [IQI]

IviRFSigGen IQ Q Offset [IQI]

IviRFSigGenIQImpairment Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure IQ Impairment [IQI]

Description

Specifies an origin offset voltage to the Q signal. The range of values allowed is -100% to +100%. The value is expressed as percentage (%).

Property Node Path

IviRFSigGen»IQ>>IQ Impairment>>IQ Q Offset [IQI]

IviRFSigGen IQ Ratio [IQI]

IviRFSigGenIQImpairment Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure IQ Impairment [IQI]

Description

Specifies the gain imbalance between the I and Q channels. For no imbalance this value is set to 0 %. The value is expressed as percentage (%).

Property Node Path

IviRFSigGen»IQ>>IQ Impairment>>IQ Ratio [IQI]

IviRFSigGen IQ Skew [IQI]

IviRFSigGenIQImpairment Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure IQ Impairment [IQI]

Description

Specifies the adjustment of the phase angle between the I and Q vectors. If this skew is zero, the phase angle is 90 degrees. The units are degrees.

Property Node Path

IviRFSigGen»IQ>>IQ Impairment>>IQ Skew [IQI]

IviRFSigGen IQ Source [MIQ]

IviRFSigGenModulateIQ Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure IQ [MIQ]

Description

Specifies the source of the signal that the signal generator uses for IQ modulation.

Property Node Path

IviRFSigGen»IQ>>IQ Source [MIQ]

Defined Values

IVIRFSIGGEN_VAL_IQ_SOURCE_DIGITAL_MODULATION_BASE

IVIRFSIGGEN_VAL_IQ_SOURCE_CDMA_BASE

IVIRFSIGGEN_VAL_IQ_SOURCE_TDMA_BASE

IVIRFSIGGEN_VAL_IQ_SOURCE_ARB_GENERATOR

IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL

IviRFSigGen IQSwap Enabled [MIQ]

IviRFSigGenModulateIQ Property

Data type	Access	Applies to	Coercion	High Level VIs
 TFI	R/W	N/A	None	IviRFSigGen Configure IQ [MIQ]

Description

Enables or disables the inverse phase rotation of the IQ signal by swapping the I and Q inputs. If VI_TRUE, the RF signal generator applies non-inverse phase rotation of the IQ signal. If VI_FALSE, the RF signal generator applies inverse phase rotation of the IQ signal.

Property Node Path

IviRFSigGen»IQ>>IQSwap Enabled [MIQ]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Power Sweep Stop [PSW]

IviRFSigGenPowerSweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Sweep Start Stop [PSW]

Description

Specifies the stop power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are in dBm.

Property Node Path

IviRFSigGen»Sweep>>Power Sweep>>Power Sweep Stop [PSW]

IviRFSigGen LF Generator Count [LF]

IviRFSigGenLFGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Specifies the number of LF generator sources available for a particular instrument.

Property Node Path

IviRFSigGen»LF Generator>>LF Generator Count [LF]

IviRFSigGen LF Generator Frequency [LF]

IviRFSigGenLFGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure LF Generator [LF]

Description

Specifies the frequency of the active LF generator. The unit is Hz.

Property Node Path

IviRFSigGen»LF Generator>>LF Generator Frequency [LF]

IviRFSigGen LF Generator Output Amplitude [LFO]

IviRFSigGenLFGeneratorOutput Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure LF Generator Output [LFO]

Description

Specifies the output voltage of the LF generator. The unit is V.

Property Node Path

IviRFSigGen»LF Generator>>LF Generator Outputs>>LF Generator
Output Amplitude [LFO]

IviRFSigGen LF Generator Output Enabled [LFO]

IviRFSigGenLFGeneratorOutput Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure LF Generator Output [LFO]

Description

Specifies whether the LF generator applies an output signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»LF Generator>>LF Generator Outputs>>LF Generator
Output Enabled [LFO]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen LF Generator Waveform [LF]

IviRFSigGenLFGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure LF Generator [LF]

Description

Specifies the waveform of the active LF generator.

Property Node Path

IviRFSigGen»LF Generator>>LF Generator Waveform [LF]

Defined Values

IVIRFSIGGEN VAL LF GENERATOR WAVEFORM SINE

IVIRFSIGGEN VAL LF GENERATOR WAVEFORM SQUARE

IVIRFSIGGEN VAL LF GENERATOR WAVEFORM TRIANGLE

IVIRFSIGGEN VAL LF GENERATOR WAVEFORM RAMP UP

IVIRFSIGGEN VAL LF GENERATOR WAVEFORM RAMP DOWN

IviRFSigGen List Dwell [LST]

IviRFSigGenList Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure List Dwell [LST]

Description

Specifies the duration time of one step. The units are in seconds.
This property is ignored if List Single Step Enabled is set to VI_TRUE.

Property Node Path

IviRFSigGen»Sweep>>List>>List Dwell [LST]

IviRFSigGen List Selected Name [LST]

IviRFSigGenList Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select List [LST]

Description

Specifies the name of the selected list to become active. The name must be one of the lists created .

Property Node Path

IviRFSigGen»Sweep>>List>>List Selected Name [LST]

IviRFSigGen List Single Step Enabled [LST]

IviRFSigGenList Property

Data type	Access	Applies to	Coercion	High Level VIs
 Boolean	R/W	N/A	None	IviRFSigGen Configure List Dwell [LST]

Description

Enables or disables single step mode.

VI_TRUE: The list will advance when the next trigger event occurs.

VI_FALSE: The list will advance immediatly after the dwell time ends.

Property Node Path

IviRFSigGen»Sweep>>List>>List Single Step Enabled [LST]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Modulation Source Count [MS]

IviRFSigGenAnalogModulationSource Property

Data type	Access	Applies to	Coercion	High Level VIs
I32I	RO	N/A	None	None

Description

Specifies how many analog modulation sources are available.

Property Node Path

IviRFSigGen»Analog Modulation>>Source>>Modulation Source Count
[MS]

IviRFSigGen Output Enabled

IviRFSigGenBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Output Enabled

Description

Specifies whether to enable or disable the RF output signal.

Property Node Path

IviRFSigGen»RF>>Output Enabled

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen PM Deviation [PM]

IviRFSigGenModulatePM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure PM [PM]

Description

Specifies the extent of modulation (peak phase deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal. The units are radians.

Property Node Path

IviRFSigGen»Analog Modulation>>PM>>PM Deviation [PM]

IviRFSigGen PM Enabled [PM]

IviRFSigGenModulatePM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	<u>IviRFSigGen Configure PM Enabled [PM]</u>

Description

Specifies whether the signal generator applies phase modulation to the RF output signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»Analog Modulation>>PM>>PM Enabled [PM]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen PM External Coupling [PM]

IviRFSigGenModulatePM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure PM External Coupling [PM]

Description

Specifies the coupling of the external source of the modulating signal.

Property Node Path

IviRFSigGen»Analog Modulation>>PM>>PM External Coupling [PM]

Defined Values

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_AC

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_DC

IviRFSigGen PM Nominal Voltage [PM]

IviRFSigGenModulatePM Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	None

Description

This is the voltage at which the instrument achieves the amount of modulation specified by the [IviRFSigGen>>Analog Modulation>>PM>>PM Deviation \[PM\]](#) property.

Note: This property is read only.

Property Node Path

IviRFSigGen»Analog Modulation>>PM>>PM Nominal Voltage [PM]

IviRFSigGen PM Source [PM]

IviRFSigGenModulatePM Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure PM [PM]

Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas.

Property Node Path

IviRFSigGen»Analog Modulation>>PM>>PM Source [PM]

IviRFSigGen Power Level

IviRFSigGenBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure RF

Description

Specifies the amplitude (power level) of the RF output signal. The value is in dBm.

Property Node Path

IviRFSigGen»RF>>Power Level

IviRFSigGen Power Step Dwell [PST]

IviRFSigGenPowerStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Step Dwell [PST]

Description

Specifies the duration time of one step. The units are in seconds.
This property is ignored if Frequency Step Single Step Enabled is set to VI_TRUE.

Property Node Path

IviRFSigGen»Sweep>>Power Step>>Power Step Dwell [PST]

IviRFSigGen Power Step Size [PST]

IviRFSigGenPowerStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Step Start Stop [PST]

Description

Specifies the step size. The units are in dBm.

Property Node Path

IviRFSigGen»Sweep»»Power Step»»Power Step Size [PST]

IviRFSigGen Power Step Start [PST]

IviRFSigGenPowerStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Step Start Stop [PST]

Description

Specifies the start power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are in dBm.

Property Node Path

IviRFSigGen»Sweep>>Power Step>>Power Step Start [PST]

IviRFSigGen Power Step Stop [PST]

IviRFSigGenPowerStep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Step Start Stop [PST]

Description

Specifies the stop power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are in dBm.

Property Node Path

IviRFSigGen»Sweep>>Power Step>>Power Step Stop [PST]

IviRFSigGen Power Sweep Start [PSW]

IviRFSigGenPowerSweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Power Sweep Start Stop [PSW]

Description

Specifies the start power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are in dBm.

Property Node Path

IviRFSigGen»Sweep>>Power Sweep>>Power Sweep Start [PSW]

IviRFSigGen Power Sweep Time [PSW]

IviRFSigGenPowerSweep Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	<u>IviRFSigGen Configure Power Sweep Time [PSW]</u>

Description

Specifies the duration of one sweep from start to stop power. The units are in seconds.

Property Node Path

IviRFSigGen»Sweep>>Power Sweep>>Power Sweep Time [PSW]

IviRFSigGen Pulse Double Delay [DPG]

IviRFSigGenPulseDoubleGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Pulse Double [DPG]

Description

Specifies the delay of the second pulse. The units are in seconds.

Property Node Path

IviRFSigGen»Pulse Generator>>Double Pulse Generators>>Pulse
Double Delay [DPG]

IviRFSigGen Pulse Double Enabled [DPG]

IviRFSigGenPulseDoubleGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Double [DPG]

Description

Specifies whether double pulse mode is used (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»Pulse Generator>>Double Pulse Generators>>Pulse
Double Enabled [DPG]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Pulse External Trigger Delay [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Pulse External Trigger [PG]

Description

Specifies the delay for starting the output pulse with respect to the trigger input. The units are in seconds.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse External Trigger Delay [PG]

IviRFSigGen Pulse External Trigger Slope [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse External Trigger [PG]

Description

Specifies whether the event occurs on the rising or falling edge of the input signal.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse External Trigger Slope [PG]

Defined Values

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_POSITIVE

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_NEGATIVE

IviRFSigGen Pulse Gating Enabled [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse [PG]

Description

Specifies whether pulse gating is enabled(VI_TRUE) or disabled(VI_FALSE).

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Gating Enabled [PG]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Pulse Internal Trigger Period [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Pulse Internal Trigger [PG]

Description

Specifies the period of the pulse generators output signal (if Pulse Trigger Source is set to Internal). The units are in seconds.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Internal Trigger Period [PG]

IviRFSigGen Pulse Modulation Enabled [PULM]

IviRFSigGenModulatePulse Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Modulation Enabled [PULM]

Description

Specifies whether the signal generator applies pulse modulation to the RF output signal (VI_TRUE) or not (VI_FALSE).

Property Node Path

IviRFSigGen»Pulse Modulation>>Pulse Modulation Enabled [PULM]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Pulse Modulation External Polarity [PULM]

IviRFSigGenModulatePulse Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Modulation External Polarity [PULM]

Description

Specifies the polarity of the external source signal.

Property Node Path

IviRFSigGen»Pulse Modulation>>Pulse Modulation External Polarity
[PULM]

Defined Values

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NO

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_IN

IviRFSigGen Pulse Modulation Source [PULM]

IviRFSigGenModulatePulse Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Modulation Source [PULM]

Description

Specifies the source of the signal that is used as the modulating signal.

Property Node Path

IviRFSigGen»Pulse Modulation>>Pulse Modulation Source [PULM]

Defined Values

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_EXTERNAL

IviRFSigGen Pulse Output Enabled [PGO]

IviRFSigGenPulseGeneratorOutput Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Output [PGO]

Description

Specifies whether the output of the pulse generator is on (VI_TRUE) or off (VI_FALSE).

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Generator Output>>Pulse Output
Enabled [PGO]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Pulse Output Polarity [PGO]

IviRFSigGenPulseGeneratorOutput Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Pulse Output [PGO]

Description

Specifies the polarity of the output signal.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Generator Output>>Pulse Output
Polarity [PGO]

Defined Values

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_NORMAL

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_INVERSE

IviRFSigGen Pulse Trigger Source [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------

	R/W	N/A	None	IviRFSigGen Configure Pulse [PG]
-----------------------------------------------------------------------------------	-----	-----	------	--------------------------------------------------

Description

Specifies the source of the signal the pulse generator uses to generate one pulse.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Trigger Source [PG]

Defined Values

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_EXTERNAL

IviRFSigGen Pulse Width [PG]

IviRFSigGenPulseGenerator Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviRFSigGen Configure Pulse [PG]

Description

Specifies the width of the output pulse. The units are in seconds.

Property Node Path

IviRFSigGen»Pulse Generator>>Pulse Width [PG]

IviRFSigGen Reference Oscillator External Frequency [RO]

IviRFSigGenReferenceOscillator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Reference Oscillator [RO]

Description

Specifies the frequency of the external signal, which is used as reference for internal RF frequency generation. This value is used only if the [IviRFSigGen>>Reference Oscillator>>Reference Oscillator Source \[RO\]](#) property is set to External.

Property Node Path

IviRFSigGen»Reference Oscillator>>Reference Oscillator External
Frequency [RO]

IviRFSigGen Reference Oscillator Source [RO]

IviRFSigGenReferenceOscillator Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Reference Oscillator [RO]

Description

Specifies the reference frequency source used to generate the exact RF output frequency.

Property Node Path

IviRFSigGen»Reference Oscillator>>Reference Oscillator Source [RO]

Defined Values

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL

IviRFSigGen Single Step Enabled [PST]

IviRFSigGenPowerStep Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Power Step Dwell [PST]

Description

Specifies whether the trigger initiates the next step (VI_TRUE), or the next step is taken after dwell time (VI_FALSE).

Property Node Path

IviRFSigGen»Sweep>>Power Step>>Single Step Enabled [PST]

Defined Values

VI_TRUE

VI_FALSE

IviRFSigGen Sweep Mode [SWP]

IviRFSigGenSweep Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Sweep

Description

Specifies the sweep mode applied to the output signal.

Property Node Path

IviRFSigGen»Sweep>>Sweep Mode [SWP]

Defined Values

IVIRFSIGGEN_VAL_SWEEP_MODE_NONE

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_SWEEP

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_SWEEP

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_STEP

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_STEP

IVIRFSIGGEN_VAL_SWEEP_MODE_LIST

IviRFSigGen Sweep Trigger Source [SWP]

IviRFSigGenSweep Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure Sweep

Description

Specifies the trigger used to start a sweep operation.

Property Node Path

IviRFSigGen»Sweep>>Sweep Trigger Source [SWP]

Defined Values

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_IMMEDIATE

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SOFTWARE

IviRFSigGen TDMA Clock Source [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure TDMA Clock Source [TDMA]

Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

Property Node Path

IviRFSigGen»TDMA>>TDMA Clock Source [TDMA]

Defined Values

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INTERNAL

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_EXTERNAL

IviRFSigGen TDMA Clock Type [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure TDMA Clock Source [TDMA]

Description

Specifies the type of the external clock signal used to generate the digital modulation. This value is used only if the [IviRFSigGen>>TDMA>>TDMA Clock Source \[TDMA\]](#) property is set to External.

Property Node Path

IviRFSigGen»TDMA>>TDMA Clock Type [TDMA]

Defined Values

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_BIT

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SYMBOL

IviRFSigGen TDMA External Trigger Slope [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure TDMA External Trigger Slope [TDMA]

Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

Property Node Path

IviRFSigGen»TDMA>>Trigger>>TDMA External Trigger Slope [TDMA]

Defined Values

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE

IviRFSigGen TDMA Frame Count [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Specifies the number of TDMA frames available for a particular instrument.

Property Node Path

IviRFSigGen»TDMA>>TDMA Frame Count [TDMA]

IviRFSigGen TDMA Selected Frame [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select TDMA Frame [TDMA]

Description

Specifies the actual frame used by the instrument. It is selected from the list queried with the Get TDMA Frame Names VI.

Property Node Path

IviRFSigGen»TDMA>>TDMA Selected Frame [TDMA]

IviRFSigGen TDMA Selected Standard [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Select TDMA Standard [TDMA]

Description

Specifies the actual standard used by the instrument. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

Property Node Path

IviRFSigGen»TDMA>>TDMA Selected Standard [TDMA]

IviRFSigGen TDMA Standard Count [TDMA]

Data type	Access	Applies to	Coercion	High Level VIs
132	RO	N/A	None	None

Description

Specifies the number of TDMA standards available for a particular instrument.

Property Node Path

IviRFSigGen»TDMA>>TDMA Standard Count [TDMA]

IviRFSigGen TDMA Trigger Source [TDMA]

IviRFSigGenTDMABase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviRFSigGen Configure TDMA Trigger Source [TDMA]

Description

Specifies the source of the trigger signal that starts the frame/slots generation.

Property Node Path

IviRFSigGen»TDMA>>Trigger>>TDMA Trigger Source [TDMA]

Defined Values

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_IMMEDIATE

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_EXTERNAL

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SOFTWARE

IviRFSigGen Defined Values

Expand this book to view an alphabetized list of IviRFSigGen defined values.

IVIRFSIGGEN_VAL_ALC_SOURCE_EXTERNAL

Description

The ALC is controlled by an external voltage.

Defined Value

2

IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL

Description

The ALC is controlled by an internal measurement source.

Defined Value

1

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_

Description

The external source is coupled for AC only.

Defined Value

1

IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_

Description

The external source is coupled for both DC and AC

Defined Value

2

IVIRFSIGGEN_VAL_AM_SCALING_LINEAR

Description

Enables linear attenuation for amplitude modulation.

Defined Value

1

IVIRFSIGGEN_VAL_AM_SCALING_LOGARITHMIC

Description

Enables logarithmic attenuation for amplitude modulation.

Defined Value

2

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_

Description

Enables falling edge triggering.

Defined Value

2

IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_

Description

Enables rising edge triggering.

Defined Value

1

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_EX

Description

The sweep is started with an external signal.

Defined Value

2

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IM

Description

The ARB generator system does not wait for a trigger. The ARB runs continuously.

Defined Value

1

IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SC

Description

The sweep is started with a software programmable trigger.

Defined Value

3

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_EX

Description

A connected external clock generator (bit or symbol clock frequency) is used.

Defined Value

2

IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INT

Description

The internal clock generator is used.

Defined Value

1

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGEF

Description

Enables falling edge triggering.

Defined Value

2

IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGEF

Description

Enables rising edge triggering.

Defined Value

1

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_I

Description

Each channel coding is started with an external signal.

Defined Value

2

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_I

Description

The CDMA generator system does not wait for a trigger. Each channel coding is run continuously.

Defined Value

1

IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_S

Description

Each channel coding is started with a software programmable trigger.

Defined Value

3

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

A connected external clock generator (bit or symbol clock frequency) is used.

Defined Value

2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

An internal clock generator is used.

Defined Value

1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

A constant bit sequence is used as data source and repeated continuously.

Defined Value

3

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

The data from an external device connected to the instrument is used.

Defined Value

1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

The internal PRBS (Pseudo Random Binary Sequence) generator is used as data source.

Defined Value

2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

The external clock frequency is equal to the bit clock frequency of the digital modulation.

Defined Value

1

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

The external clock frequency is equal to the symbol clock frequency of the digital modulation.

Defined Value

2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{11} - 1$.

Defined Value

2

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{15} - 1$.

Defined Value

3

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{16} - 1$.

Defined Value

4

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{20} - 1$.

Defined Value

5

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{21} - 1$.

Defined Value

6

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^{23} - 1$.

Defined Value

7

IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BAS

Description

Length of PRBS sequence is $2^9 - 1$.

Defined Value

1

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_

Description

The external source is coupled for AC only.

Defined Value

1

IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_

Description

The external source is coupled for both DC and AC.

Defined Value

2

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALIN

Description

Enables linear scaling.

Defined Value

1

IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALIN

Description

Enables logarithmic scaling.

Defined Value

2

IVIRFSIGGEN_VAL_IQ_SOURCE_ARB_GENERAT

Description

The signal generator uses the internally generated Arb signal to apply IQ modulation to the output RF signal.

Defined Value

4

IVIRFSIGGEN_VAL_IQ_SOURCE_CDMA_BASE

Description

The signal generator uses the internally generated CDMA signal to apply IQ modulation to the output RF signal.

Defined Value

2

IVIRFSIGGEN_VAL_IQ_SOURCE_DIGITAL_MODU

Description

The signal generator uses the internally generated digital modulation signal to apply IQ modulation to the output RF signal.

Defined Value

1

IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL

Description

The signal generator uses data from an external source for IQ modulation.

Defined Value

5

IVIRFSIGGEN_VAL_IQ_SOURCE_TDMA_BASE

Description

The signal generator uses the internally generated TDMA signal to apply IQ modulation to the output RF signal.

Defined Value

3

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORI

Description

Configures the LF generator to produce a falling ramp waveform.

Defined Value

5

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORI

Description

Configures the LF generator to produce a rising ramp waveform.

Defined Value

4

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORI

Description

Configures the LF generator to produce a sinusoidal waveform.

Defined Value

1

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORI

Description

Configures the LF generator to produce a square waveform.

Defined Value

2

IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORI

Description

Configures the LF generator to produce a triangle waveform.

Defined Value

3

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_

Description

The external source is coupled for AC only.

Defined Value

1

IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_

Description

The external source is coupled for both DC and AC.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGE

Description

Enables falling edge triggering.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGE

Description

Enables rising edge triggering.

Defined Value

1

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTE

Description

The signal generator modulates the carrier signal with inverted pulse polarity.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTE

Description

The signal generator modulates the carrier signal with normal pulse polarity.

Defined Value

1

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOUF

Description

An external generator is used for modulation.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_MODULATION_SOUP

Description

An internal pulse generator (IviRFSigGenPulseGenerator Extension Group) is used for modulation.

Defined Value

1

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_

Description

Specifies inverted polarity.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_

Description

Specifies normal polarity.

Defined Value

1

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_

Description

The pulse is started with a trigger after the delay time specified by the IviRFSigGen>>Pulse Generator>>Pulse External Trigger Delay [PG] property.

Defined Value

2

IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_

Description

No external trigger is used. The pulse period is specified by the IviRFSigGen>>Pulse Generator>>Pulse Internal Trigger Period [PG] property.

Defined Value

1

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_

Description

An external reference oscillator is used.

Defined Value

2

IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_

Description

An internal reference oscillator is used.

Defined Value

1

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY

Description

The signal generator sweeps the RF output signals frequency in steps.

Defined Value

4

IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY

Description

The signal generator sweeps the RF output signal's frequency in an analog form (non-stepped).

Defined Value

2

IVIRFSIGGEN_VAL_SWEEP_MODE_LIST

Description

The signal generator uses two lists with frequency and power level values to sweep the RF output signals.

Defined Value

6

IVIRFSIGGEN_VAL_SWEEP_MODE_NONE

Description

The RF output of the signal generator is a non-swept signal (Continuous Wave). Frequency and power level from base capability group is used.

Defined Value

1

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_STE

Description

The signal generator sweeps the RF output signals power level in steps.

Defined Value

5

IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_SW

Description

The signal generator sweeps the RF output signal's power level in an analog form (non-stepped).

Defined Value

3

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_

Description

The sweep is started with an external signal.

Defined Value

2

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_

Description

The sweep system does not wait for a trigger of any kind, so it is running continuously.

Defined Value

1

IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_

Description

The sweep is started with a software programmable trigger.

Defined Value

3

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_EX

Description

A connected external clock generator (bit or symbol clock frequency) is used.

Defined Value

2

IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INT

Description

The internal clock generator is used.

Defined Value

1

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_1

Description

The external clock frequency is equal to the bit clock frequency of the digital modulation.

Defined Value

1

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_1

Description

The external clock frequency is equal to the symbol clock frequency of the digital modulation.

Defined Value

2

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER

Description

Enables falling edge triggering.

Defined Value

2

IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER

Description

Enables rising edge triggering.

Defined Value

1

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_E

Description

Each frame is started with an external signal.

Defined Value

2

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_I

Description

The TDMA generator system does not wait for a trigger of any kind, so it is running the frames continuously.

Defined Value

1

IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_S

Description

Each frame is started with a software programmable trigger.

Defined Value

3

□ IviScope Class Driver Reference for LabVIEW

Concepts

[IviScope Class Driver Overview](#)

[IviScope Extension Groups](#)

[IviScope Behavior Models](#)

IviScope VI and Property Reference

[IviScope VI Tree](#)

[IviScope Properties by Group](#)

[IviScope Property Value Definitions](#)

[IviScope Error and Completion Codes](#)

Other Reference

[Interchangeability Checking](#)

[Simulation](#)

IviScope Class Driver Overview

This instrument driver provides programming support for the IviScope Class. The IviScope class driver conceptualizes an oscilloscope as an instrument that can acquire a voltage waveform from an analog input signal. The points in the waveform are acquired at a configurable interval and could be acquired sequentially in real-time sampling or interleaved from multiple waveform acquisitions in equivalent-time or RIS sampling. A property of one of the input signals (typically a rising or falling edge) is used to trigger the acquisition. In addition, the IviScope class driver can support instruments that have more complex acquisition modes (such as average, envelope, and peak detect) and trigger types (such as TV, runt, and glitch). The driver contains VIs for all the functions that the IVI Foundation requires for the IviScope specification. This driver requires NI-VISA and the IVI Compliance Package to be installed.

Use this driver to develop programs that are independent of a particular oscilloscope. You can use this class driver with any oscilloscope that has an IVI instrument specific driver that is compliant with the IviScope class. The IviScope class driver accesses the specific driver for your instrument using the configuration information you supply in MAX.

The IviScope class driver divides instrument capabilities into a base capability group and multiple extension capability groups. The [IviScopeBase capability group](#) supports oscilloscopes that acquire one waveform after receiving a trigger signal. The VIs and properties of this group are implemented by all oscilloscope instrument drivers that are compliant with the IviScope class.

Extension Capability Groups

The IviScope class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Abbreviation	Description
<u>IviScopeTVTrigger</u>	[TV]	For instruments capable of triggering on standard television signals
<u>IviScopeRuntTrigger</u>	[RT]	For instruments capable of triggering on runts
<u>IviScopeGlitchTrigger</u>	[GT]	For instruments capable of triggering on glitches
<u>IviScopeWidthTrigger</u>	[WT]	For instruments capable of triggering on a variety of conditions regarding pulse widths
<u>IviScopeWaveformMeasurement</u>	[WM]	For instruments capable of calculating measurements from an acquired waveform, such as rise time or frequency
<u>IviScopeMinMaxWaveform</u>	[MM]	For instruments capable of acquiring minimum and maximum waveforms that correspond to the same range of time
<u>IviScopeInterpolation</u>	[I]	For instruments capable of configuring the acquisition to

		interpolate missing points in a waveform
<u>IviScopeAcLineTrigger</u>	[AT]	For instruments capable of triggering on zero crossings of a network supply voltage
<u>IviScopeProbeAutoSense</u>	[PAS]	For instruments capable of automatically sensing the probe attenuation of an attached probe
<u>IviScopeContinuousAcquisition</u>	[CA]	For instruments capable of continuously acquiring data from the input and displaying it on the screen
<u>IviScopeAverageAcquisition</u>	[AA]	For instruments capable of creating a waveform that is the average of multiple waveform acquisitions
<u>IviScopeSampleMode</u>	[SM]	For instruments capable of returning the actual sample mode
<u>IviScopeTriggerModifier</u>	[TM]	For instruments capable of modifying the behavior of the triggering subsystem in the absence of an expected trigger
<u>IviScopeAutoSetup</u>	[AS]	For instruments with the automatic configuration ability

Related Topics

[IVI Inherent Class Capabilities](#)

IviScope Extension Capability Groups

The IviScope class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviScopeTVTrigger</u>	For instruments capable of triggering on standard television signals
<u>IviScopeRuntTrigger</u>	For instruments capable of triggering on runs
<u>IviScopeGlitchTrigger</u>	For instruments capable of triggering on glitches
<u>IviScopeWidthTrigger</u>	For instruments capable of triggering on a variety of conditions regarding pulse widths
<u>IviScopeWaveformMeasurement</u>	For instruments capable of calculating measurements from an acquired waveform, such as rise time or frequency
<u>IviScopeMinMaxWaveform</u>	For instruments capable of acquiring minimum and maximum waveforms that correspond to the same range of time
<u>IviScopeInterpolation</u>	For instruments capable of configuring the acquisition to interpolate missing points in a waveform
<u>IviScopeAcLineTrigger</u>	For instruments capable of triggering on zero crossings of a network supply voltage
<u>IviScopeProbeAutoSense</u>	For instruments capable of automatically sensing the probe attenuation of an attached probe

<u>IviScopeContinuousAcquisition</u>	For instruments capable of continuously acquiring data from the input and displaying it on the screen
<u>IviScopeAverageAcquisition</u>	For instruments capable of creating a waveform that is the average of multiple waveform acquisitions
<u>IviScopeSampleMode</u>	For instruments capable of returning the actual sample mode
<u>IviScopeTriggerModifier</u>	For instruments capable of modifying the behavior of the triggering subsystem in the absence of an expected trigger
<u>IviScopeAutoSetup</u>	For instruments with the automatic configuration ability

IviScopeAcLineTrigger Extension Group

In addition to the fundamental capabilities, the IviScopeAcLineTrigger extension group defines extensions for oscilloscopes that are capable of synchronizing the trigger with the AC Line.

AC Line triggering occurs when the oscilloscope detects a positive zero crossing, negative zero crossing, or optionally either positive or negative zero crossing on the network supply voltage.

IviScopeAcLineTrigger VI

[IviScope Configure AC Line Trigger Slope \[AT\]](#)

IviScopeAcLineTrigger Property

[IviScope AC Line Trigger Slope](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeAutoSetup Extension Group

The IviScopeAutoSetup extension group provides support for oscilloscopes that can perform an auto-setup operation.

IviScopeAutoSetup VI

[IviScope Auto Setup](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeAverageAcquisition Extension Group

The IviScopeAverageAcquisition extension group defines extensions for oscilloscopes that can perform average acquisition.

IviScopeAverageAcquisition VI

[IviScope Configure Initiate Continuous \[CA\]](#)

IviScopeAverageAcquisition Property

[IviScope Initiate Continuous](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScope Base Capability Group

The IviScopeBase capability group supports oscilloscopes that can acquire waveforms from multiple channels with an edge trigger. The IviScopeBase capability group defines properties and their values to configure the oscilloscope's channel, acquisition, and trigger sub-systems. The IviScopeBase capability group also includes VIs for configuring the oscilloscope as well as initiating waveform acquisition and retrieving waveforms. Scroll down to see all base VIs and properties.

[Ivi Inherent VIs](#)

[Ivi Inherent Properties](#)

[IviScope behavior model](#)

IviScope Sub-Systems

[Channel Sub-System](#)

[Acquisition Sub-System](#)

[Trigger Sub-System](#)

IviScopeBase VIs

[IviScope Configure Acquisition Type](#)

[IviScope Configure Channel](#)

[IviScope Configure Channel Characteristics](#)

[IviScope Configure Acquisition Record](#)

[IviScope Actual Record Length](#)

[IviScope Configure Trigger](#)

[IviScope Configure Trigger Coupling](#)

[IviScope Configure Edge Trigger Source](#)

[IviScope Read Waveform](#)

[IviScope Abort](#)

[IviScope Initiate Acquisition](#)

[IviScope Acquisition Status](#)

[IviScope Fetch Waveform](#)

[IviScope Is Invalid Waveform Element](#)

[IviScope Actual Sample Rate](#)

IviScopeBase Properties

[IviScope Acquisition Start Time](#)

[IviScope Acquisition Type](#)

[IviScope Channel Enabled](#)

[IviScope Horizontal Minimum Number of Points](#)

[IviScope Horizontal Record Length](#)

[IviScope Horizontal Sample Rate](#)

[IviScope Horizontal Time Per Record](#)

[IviScope Input Impedance](#)

[IviScope Maximum Input Frequency](#)

[IviScope Probe Attenuation](#)

[IviScope Trigger Coupling](#)

[IviScope Trigger Holdoff](#)

[IviScope Trigger Level](#)

[IviScope Trigger Slope](#)

[IviScope Trigger Source](#)

[IviScope Trigger Type](#)

[IviScope Vertical Coupling](#)

[IviScope Vertical Offset](#)

[IviScope Vertical Range](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

IviScope Base Extension Capability

Sub-Systems

Expand this topic to view base sub-system capability groups.

Channel Sub-System

The channel sub-system configures the range of voltages the oscilloscope acquires and how the oscilloscope couples the input signal to the acquisition sub-system. The main channel sub-system properties include:

- [IviScope Vertical Range](#)
- [IviScope Vertical Offset](#)
- [IviScope Vertical Coupling](#)
- [IviScope Probe Attenuation](#)
- [IviScope Channel Enabled](#)

All of the channel sub-system properties are channel-based. They can be set individually or they can be set as a group with the [IviScope Configure Channel](#) VI.



Channel Sub-System Properties

The [IviScope Vertical Range](#) property specifies the absolute value of the range of voltages that the oscilloscope acquires. The [IviScope Vertical Offset](#) property specifies the center of the range specified by the [IviScope Vertical Range](#) property with respect to ground. The [IviScope Vertical Coupling](#) property specifies how to couple the input signal to the channel sub-system. The [IviScope Probe Attenuation](#) property specifies the scaling factor by which the probe attenuates the input signal. Typically, the value of the [IviScope Probe Attenuation](#) property determines the range values the driver accepts for the [IviScope Vertical Range](#) and [IviScope Vertical Offset](#) properties. The [IviScope Channel Enabled](#) property specifies whether the oscilloscope acquires a waveform for the channel.

Trigger Sub-System

The trigger sub-system configures the type of event that triggers the oscilloscope. The global trigger sub-system properties are:

- [IviScope Trigger Type](#)
- [IviScope Trigger Holdoff](#)
- [IviScope Trigger Coupling](#)

The [IviScope Trigger Type](#) property specifies the event that triggers the oscilloscope.

The [IviScope Trigger Holdoff](#) property specifies the length of time after the oscilloscope detects a trigger during which the oscilloscope ignores additional triggers. The [IviScope Trigger Holdoff](#) property only affects instrument operation when the oscilloscope requires multiple acquisitions to build a complete waveform. The oscilloscope requires multiple waveform acquisitions when the instrument uses equivalent time sampling or the acquisition type is set to envelope or average. The [IviScope Trigger Coupling](#) property specifies how the oscilloscope couples the trigger source signal to the trigger sub-system.

The properties from the above list can be set individually or they can be set as a group with the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs.

The following properties configure the edge trigger. These properties can be set individually or as a group with the [IviScope Configure Edge Trigger Source](#) VI.

- [IviScope Trigger Source](#)
- [IviScope Trigger Level](#)
- [IviScope Trigger Slope](#)



Edge Triggers

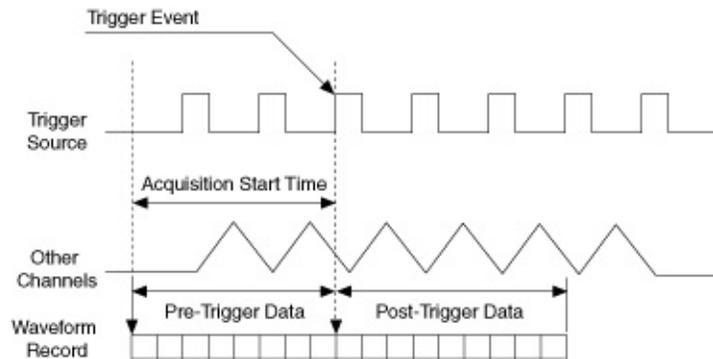
The [IviScope Trigger Source](#) property specifies the source the oscilloscope monitors for the trigger event. The [IviScope Trigger Level](#) property specifies the voltage threshold for the trigger sub-system. Most of the trigger types use the value held in the [IviScope Trigger Level](#) property.

The [IviScope Trigger Slope](#) property specifies whether a positive or negative edge triggers the oscilloscope.

When the trigger type is edge, the values held in the [IviScope Trigger Level](#) and [IviScope Trigger Slope](#) properties define the trigger event. The oscilloscope triggers when the trigger source crosses the threshold level with the polarity the [IviScope Trigger Level](#) and [IVISCOPE_ATTR_TRIGGER_SLOPE](#) properties specify.

Triggering Overview

The following figure illustrates the relationship of the Trigger event and the **acqStartTime** parameter to the data the oscilloscope acquires in the waveform record. For this illustration, the trigger type is an edge trigger and the trigger source is a series of pulses. It is important to note that the trigger source can be an analog or digital signal and the trigger type can be any trigger type that this class driver defines.



Holdoff Overview

The following figure shows how the hold off effects the trigger subsystem. Ideally the trigger event occurs at condition '1', but sometimes the oscilloscope triggers on condition '2' because the signal crosses the trigger level. When you specify a hold off, the oscilloscope triggers on conditions '1' and '3', and ignores conditions '2' and '4'.



This IviScopeBase Capability group defines VIs that retrieve waveforms from the oscilloscope. These VIs return the following information:

- The waveform record as an array of voltage.
- The time that corresponds to the first point in the waveform array relative to the Trigger Even.
- The effective time interval between points in the waveform record

Acquisition Sub-System

The acquisition sub-system configures the size of the waveform record, the length of time that corresponds to the overall waveform record, and the acquisition start time. The configurable acquisition sub-system properties include:

- [IviScope Acquisition Start Time](#)
- [IviScope Horizontal Minimum Number of Points](#)
- [IviScope Horizontal Time Per Record](#)

You specify the minimum number of points they require the oscilloscope to acquire with the [IviScope Horizontal Minimum Number of Points](#) property. The [IviScope Horizontal Time Per Record](#) property specifies the length of time that corresponds to the overall waveform record. You specify the length of time from the trigger event to the first point in the waveform record with the [IviScope Acquisition Start Time](#). If the value is positive, the first point in the waveform record occurs *after* the trigger event.

All of the acquisition properties can be set individually or they can be set as a group with the [IviScope Configure Acquisition Record VI](#).

In addition, the acquisition sub-system includes two read-only properties:

- [IviScope Horizontal Record Length](#)
- [IviScope Horizontal Sample Rate](#)

The [IviScope Horizontal Record Length](#) property returns the actual number of points in the waveform record. The [IviScope Horizontal Sample Rate](#) property returns the effective sample rate of the oscilloscope.

The IviScope class driver defines separate properties for the minimum record size you request and the actual record length. Typically, oscilloscopes change the record length dynamically when the acquisition type changes. For example, when you change the acquisition type from normal to envelope, many oscilloscopes reduce the record length by half. When you change the acquisition type, the instrument driver uses the value held in the [IviScope Horizontal Minimum Number of Points](#) property to check that the new record length is equal to or greater than the minimum record length you require.

IviScopeContinuousAcquisition Extension Group

The IviScopeContinuousAcquisition extension group provides support for oscilloscopes that can perform a continuous acquisition.

IviScopeContinuousAcquisition VI

[IviScope Configure Initiate Continuous \[CA\]](#)

IviScopeContinuousAcquisition Property

[IviScope Initiate Continuous](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

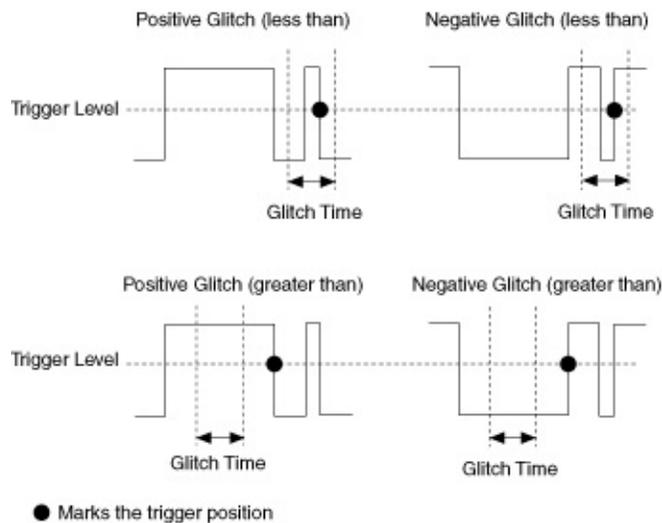
[IviScope behavior model](#)

[IviScopeContinuousAcquisition behavior model](#)

IviScopeGlitchTrigger Extension Group

In addition to the base capabilities, the IviScopeGlitchTrigger extension group defines extensions for oscilloscopes that can trigger on glitch pulses.

A glitch occurs when the oscilloscope detects a pulse width that is less than or greater than a specified glitch time. The figure below shows both positive and negative glitches, both less than and greater than a specified glitch time.



Glitch Triggers

With the IviScopeGlitchTrigger extension group you can select whether a positive glitch, negative glitch, or either triggers the acquisition. You can also select the glitch condition—whether you want the oscilloscope to trigger on pulses less than or greater than the specified glitch width.

IviScopeGlitchTrigger VIs

[IviScope Configure Glitch Trigger Source](#)

IviScopeGlitchTrigger Properties

[IviScope Glitch Condition \[GT\]](#)

[IviScope Glitch Polarity \[GT\]](#)

[IviScope Glitch Width \[GT\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeInterpolation Extension Group

The IviScopeInterpolation extension group defines extensions for oscilloscopes capable of interpolating values in the waveform record that the oscilloscope's acquisition subsystem was unable to digitize.

IviScopeInterpolation VI

[IviScope Configure Interpolation \[I\]](#)

IviScopeInterpolation Property

[IviScope Interpolation](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeMinMaxWaveform Extension Group

The IviScopeMinMaxWaveform extension group provides support for oscilloscopes that can acquire minimum and maximum waveforms that correspond to the same range of time. The two most common acquisition types in which oscilloscopes return minimum and maximum waveforms are envelope and peak detect.

IviScopeMinMaxWaveform VIs

[IviScope Read Min Max Waveform](#)

[IviScope Configure Number of Envelopes](#)

[IviScope Fetch Min Max Waveform](#)

IviScopeMinMaxWaveform Property

[IviScope Number of Envelopes \[MmW\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeProbeAutoSense Extension Group

The IviScopeProbeAutoSense extension group provides support for oscilloscopes that can sense and return the probe attenuation of the attached probe.

IviScopeProbeAutoSense VI

IviScope Auto Probe Sense Value

IviScopeProbeAutoSense Property

[IviScope Probe Sense Value](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeRuntTrigger Extension Group

In addition to the base capabilities, the IviScopeRuntTrigger extension group defines extensions for oscilloscopes with the capability to trigger on runt pulses.

A runt condition occurs when the oscilloscope detects a positive or negative going pulse that crosses one voltage threshold but fails to cross a second threshold before re-crossing the first. The figure below shows both positive and negative runt polarities.



Runt Triggers

With the IviScopeRuntTrigger extension group you can select whether a positive runt, negative runt, or either triggers the acquisition.

IviScopeRuntTrigger VIs

[IviScope Configure Runt Trigger Source](#)

IviScopeRuntTrigger Properties

[IviScope Runt High Threshold \[RT\]](#)

[IviScope Runt Low Threshold \[RT\]](#)

[IviScope Runt Polarity \[RT\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScope SampleMode Extension Group

The IviScopeSampleMode extension group provides support for oscilloscopes that can return whether they are using equivalent-time or real-time sampling to acquire waveforms.

IviScopeSampleMode VI

[IviScope Actual Sample Mode](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope Behavior Model](#)

IviScopeTriggerModifier Extension Group

The IviScopeTriggerModifier extension group provides support for oscilloscopes that can specify the behavior of the triggering subsystem in the absence of the configured trigger.

IviScopeTriggerModifier VI

[IviScope Configure Trigger Modifier](#)

IviScopeTriggerModifier Property

[IviScope Trigger Modifier \[TM\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeTVTrigger Extension Group

In addition to the base capabilities, the IviScopeTVTrigger extension group defines extensions for oscilloscopes capable of triggering on standard TV signals.

IviScopeTVTrigger VIs

[IviScope Configure TV Trigger Source](#)

[IviScope Configure TV Trigger Line Number](#)

IviScopeTVTrigger Properties

[IviScope TV Trigger Event](#)

[IviScope TV Trigger Line Number \[TV\]](#)

[IviScope TV Trigger Polarity \[TV\]](#)

[IviScope TV Trigger Signal Format \[TV\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeWaveformMeasurement Extension Group Overview

The IviScopeWaveformMeasurement extension group defines extensions for oscilloscopes capable of calculating various measurements such as rise-time, fall-time, period, and frequency from an acquired waveform.



Note In the following table, the literal string `IVISCOPE_VAL_` precedes the measurement name.

Waveform Measurement Descriptions

Measurement Name		Description
<input type="checkbox"/>	RISE_TIME	The length of time for the first rising edge of the signal to rise from the low reference level to the high reference level.
<input type="checkbox"/>	FALL_TIME	The length of time for the first falling edge of the signal to fall from the high reference level to the low reference level.
<input type="checkbox"/>	FREQUENCY	The frequency of one complete cycle in the waveform.
<input type="checkbox"/>	PERIOD	The length of time of one complete cycle in the waveform.
<input type="checkbox"/>	VOLTAGE_RMS	The true Root Mean Square voltage of the entire waveform.
<input type="checkbox"/>	VOLTAGE_CYCLE_RMS	The true Root Mean Square voltage over the first cycle in the waveform.
<input type="checkbox"/>	VOLTAGE_MAX	The maximum amplitude found in the entire waveform.
<input type="checkbox"/>	VOLTAGE_MIN	The minimum amplitude found in the entire waveform.
<input type="checkbox"/>	VOLTAGE_PEAK_TO_PEAK	The absolute difference between the VOLTAGE_MAX and the VOLTAGE_MIN in volts.
<input type="checkbox"/>	VOLTAGE_HIGH	The voltage that corresponds to 100% when using the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the maximum value found. The histogram method uses the most common value found above the middle of the waveform.

<input type="checkbox"/>	VOLTAGE_LOW	The voltage that corresponds to 0% when using the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the minimum value found. The histogram method uses the most common value found below the middle of the waveform.
<input type="checkbox"/>	VOLTAGE_AVERAGE	The arithmetic average in volts measured over the entire waveform.
<input type="checkbox"/>	VOLTAGE_CYCLE_AVERAGE	The arithmetic average in volts over the first cycle in the waveform.
<input type="checkbox"/>	WIDTH_NEG	The length of time between the mid reference level points of the first negative pulse in the waveform.
<input type="checkbox"/>	WIDTH_POS	The length of time between the mid reference level points of the first positive pulse in the waveform.
<input type="checkbox"/>	DUTY_CYCLE_NEG	The ratio of the WIDTH_NEG to the PERIOD of the first cycle in the waveform expressed as a percentage.
<input type="checkbox"/>	DUTY_CYCLE_POS	The ratio of the WIDTH_POS width to the PERIOD of the first cycle in the waveform expressed as a percentage.
<input type="checkbox"/>	AMPLITUDE	The VOLTAGE_HIGH less the VOLTAGE_LOW in volts over the entire waveform. AMPLITUDE = VOLTAGE_HIGH - VOLTAGE_LOW
	OVERSHOOT	The relative waveform distortion which follows an edge transition. It is calculated using the following

		<p>formula:</p> <ul style="list-style-type: none"> • for the rising edge: <p>where the local maximum is the maximum voltage of the signal in the first half of the time period that commences when the rising edge crosses the high reference level and concludes when the subsequent falling edge crosses the high reference level.</p> • for the falling edge: where the local minimum is the minimum value of the signal measured in the first half of the time period that commences when the falling edge crosses the low reference level and concludes when the subsequent rising edge crosses the low reference level. <p>The instrument makes the measurement on the edge closest to the beginning of the waveform record. The units are the percentage of the signal amplitude.</p>
	<p>PRESHOOT</p>	<p>The relative waveform distortion which precedes an edge transition. It is calculated using the following formula:</p> <ul style="list-style-type: none"> • for the rising edge: <p>where the local minimum is</p>

the minimum value of the signal measured in the second half of the time period that commences when the preceding falling edge crosses the low reference level and concludes when the rising edge crosses the low reference level.

- for the falling edge:

where the local maximum is the maximum voltage of the signal in the second half of the time period that commences when the preceding rising edge crosses the high reference level and concludes when the falling edge crosses the high reference level.

The instrument makes the measurement on the edge closest to the beginning of the waveform record. The units are the percentage of the signal amplitude.

IviScopeWaveformMeasurement Extension Group

The IviScopeWaveformMeasurement extension group defines a set of [waveform measurement descriptions](#).

Refer to the [behavior model](#) for this extension group.

IviScopeWaveformMeasurement VIs

[IviScope Configure Reference Levels](#)

[IviScope Read Waveform Measurement](#)

[IviScope Fetch Waveform Measurement](#)

IviScopeMeasWaveform Properties

[IviScope Measurement High Reference \[WM\]](#)

[IviScope Measurement Low Reference \[WM\]](#)

[IviScope Measurement Mid Reference \[WM\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

IviScopeWidthTrigger Extension Group

This group allows you to set [width triggering](#).

IviScopeWidthTrigger VI

[IviScope Configure Width Trigger Source](#)

IviScopeWidthTrigger Properties

[IviScope Width Condition \[WT\]](#)

[IviScope Width High Threshold \[WT\]](#)

[IviScope Width Low Threshold \[WT\]](#)

[IviScope Width Polarity \[WT\]](#)

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

[IviScope behavior model](#)

IviScopeWidthTrigger Extension Group Overview

In addition to the base capabilities, the IviScopeWidthTrigger extension group defines extensions for oscilloscopes capable of triggering on user-specified pulse widths.

Width triggering occurs when the oscilloscope detects a positive or negative pulse with a width within, or optionally outside, thresholds specified by you. The figure below shows positive and negative pulses that fall within the thresholds specified by you.



Width Triggers Within the Thresholds

The figure below shows positive and negative pulses that are not within the user-specified thresholds.



Width Triggers Outside the Thresholds

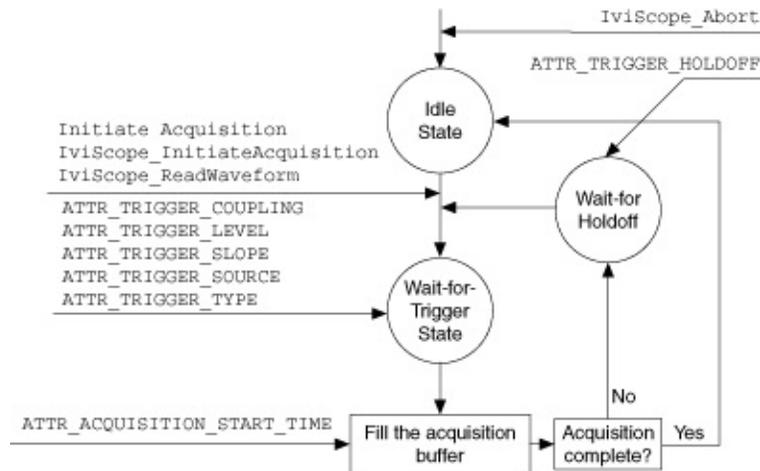
IviScope Behavior Models

[IviScope](#)

[ContinuousAcquisition](#)

IviScope Behavior Model

The following state diagram shows relationships between the IviScope Base capability group and oscilloscope behavior.



IviScope Behavior Model



Note The following extension groups follow the IviScope behavior model: [IviScopeTVTrigger](#), [IviScopeRuntTrigger](#), [IviScopeGlitchTrigger](#), [IviScopeWidthTrigger](#), [IviScopeWaveformMeasurement](#), [IviScopeProbeAutoSense](#), [IviScopeSampleMode](#), [IviScopeAcLineTrigger](#), [IviScopeInterpolation](#), [IviScopeAverageAcquisition](#), [IviScopeTriggerModifier](#), [IviScopeAutoSetup](#), and [IviScopeMinMaxWaveform](#).

Typically, you configure the oscilloscope while it is in the Idle state. You can configure the oscilloscope by accessing the IviScope properties individually or by calling the high-level Configure Channel, Configure Acquisition Record, Configure Trigger, and Configure Edge Trigger Source VIs.

To acquire waveforms, the IviScope class driver presents the high-level Read Waveform VI, as well as the low-level VIs Initiate Acquisition, Acquisition Complete, Fetch Waveform, and Abort.

The Read Waveform VI initiates a waveform acquisition and returns the acquired waveform after the oscilloscope has returned to the Idle state.

The Initiate Acquisition, Fetch Waveform, and Abort VIs give you low-level control over the measurement process. Initiate Acquisition VI

initiates a waveform acquisition and moves the instrument into the Wait-For-Trigger state. You configure the type of trigger with the trigger sub-system properties or with the Configure Trigger and Configure Edge Trigger Source VIs.

After the trigger event occurs, the oscilloscope acquires a waveform based on the channel and acquisition sub-system properties. If the oscilloscope was able to fill all of the points in the waveform in real-time it then returns to the Idle state. However, if the oscilloscope must acquire multiple waveforms in equivalent-time sampling to build up the waveform record, it then moves to the Wait-For-Holdoff state. The oscilloscope then waits configurable amount of time before moving to the Wait-For-Trigger state.

After enough waveforms have been acquired to fill the waveform record, the oscilloscope returns to the Idle state. You can use the Acquisition Status VI to determine if the acquisition is complete or is still in progress.

You can use the Fetch Waveform VI to return a waveform from a previously initiated acquisition. The Read Waveform and Fetch Waveform VIs have the following outputs:

- A waveform array
- The time of the first point in the waveform array relative to the trigger event
- The effective time interval between points in the array

IviScope ContinuousAcquisition Behavior Model

The following behavior diagram shows relationships between IviScopeContinuousAcquisition capabilities and oscilloscope behavior.



IviScope ContinuousAcquisition Behavior Model

The IviScopeContinuousAcquisition extension group adds the property that controls whether the instrument operates in a single-shot mode or it acquires the data continuously.

After the oscilloscope completes an acquisition, if the [IviScope Initiate Continuous \[CA\]](#) property is set to TRUE, the instrument goes to the Wait-for-Trigger state instead of returning to the Idle state. In the Wait-for-Trigger state, the oscilloscope display updates continuously. This specification does not define the behavior of the read and fetch VIs when this property is set to TRUE. The behavior of these VIs is instrument specific.

IviScope VI Tree

The VI tree for the IviScope class driver, including IVI and VXI*plug&play* required VIs, is shown below.



Note You use the LabVIEW Property Node to get and set properties.

IviScope VI Tree

Name or Class	VI Name	Required By
Initialize	IviScope Initialize	VPP
Initialize with Options	IviScope Initialize With Options	IVI
Configuration		
Auto Setup [AS]	IviScope Auto Setup	IviScopeAutoSetup
Acquisition Subsystem		
Configure Acquisition Type	IviScope Configure Acquisition Type	IviScope
Configure Acquisition Record	IviScope Configure Acquisition Record	IviScope
Configure Number of Averages	IviScope Configure Number of Averages	IviScopeAverageAcquisition
Configure Number of Envelopes [MnW]	IviScope Configure Number of Envelopes	IviScopeMinMaxWaveform
Configure Interpolation [I]	IviScope Configure Interpolation	IviScopeInterpolation
Configure Initiate Continuous [CA]	IviScope Configure Initiate Continuous	IviScopeContinuousAcquisition
Channel		
Configure Channel	IviScope Configure Channel	IviScope
Configure	IviScope Configure	IviScope

Channel
Characteristics

[Channel](#)
[Characteristics](#)

Trigger

Configure Trigger [IviScope Configure Trigger](#) IviScope

Configure Trigger Coupling [IviScope Configure Trigger Coupling](#) IviScope

Configure Trigger Modifier [TM] [IviScope Configure Trigger Modifier](#) IviScopeTriggerModifier

Configure Edge Trigger Source [IviScope Configure Edge Trigger Source](#) IviScope

Configure TV Trigger Source [TV] [IviScope Configure TV Trigger Source](#) IviScopeTVTrigger

Configure TV Trigger Line Number [TV] [IviScope Configure TV Trigger Line Number](#) IviScopeTVTrigger

Configure Runt Trigger Source [RT] [IviScope Configure Runt Trigger Source](#) IviScopeRuntTrigger

Configure Glitch Trigger Source [GT] [IviScope Configure Glitch Trigger Source](#) IviScopeGlitchTrigger

Configure Width Trigger Source [WT] [IviScope Configure Width Trigger Source](#) IviScopeWidthTrigger

Configure AC Line Trigger Slope [AT] [IviScope Configure AC Line Trigger Slope](#) IviScopeAcLineTrigger

Measurement

Configure Reference Levels	<u>IviScope Configure Reference Levels</u>	IviScopeWaveformMeas
----------------------------	------------------------------------------------------------	----------------------

Configuration Information

Actual Record Length	<u>IviScope Actual Record Length</u>	IviScope
Auto Probe Sense Value [PAS]	<u>IviScope Auto Probe Sense Value</u>	IviScopeProbeAutoSense
Actual Sample Mode [SM]	<u>IviScope Actual Sample Mode</u>	IviScopeSampleMode
Actual Sample Rate	<u>IviScope Sample Rate</u>	IviScope

Waveform Acquisition

Read Waveform	<u>IviScope Read Waveform</u>	IviScope
Read Min/Max Waveform	<u>IviScope Read Min Max Waveform</u>	IviScopeMinMaxWaveform
Read Waveform Measurement	<u>IviScope Read Waveform Measurement</u>	IviScopeWaveformMeas

Low-Level Acquisition

Abort	<u>IviScope Abort</u>	IviScope
Acquisition Status	<u>IviScope Acquisition Status</u>	IviScope
Fetch Min/Max Waveform [MmW]	<u>IviScope Fetch Min Max Waveform</u>	MinMaxWaveform
Fetch Waveform	<u>IviScope Fetch</u>	IviScope

	<u>Waveform</u>	
Fetch Waveform Measurement [WM]	<u>IviScope Fetch Waveform Measurement</u>	IviScopeWaveformMeas
Initiate Acquisition	<u>IviScope Initiate Acquisition</u>	IviScope
Utility		
Reset	<u>IviScope Reset</u>	VPP
Reset With Defaults	<u>Reset With Defaults</u>	VPP
Self-Test	<u>IviScope Self-Test</u>	VPP
Disable	<u>Disable</u>	VPP
Invalidate All Attributes	<u>Invalidate All Attributes</u>	VPP
Revision Query	<u>IviScope Revision Query</u>	VPP
Error-Query	<u>IviScope Error-Query</u>	VPP
Error Message	<u>IviScope Error Message</u>	VPP
Get Channel Name	<u>Get Channel Name</u>	VPP
Is Invalid Waveform Element	<u>IviScope Is Invalid Waveform Element</u>	IviScope
Interchangeability Info		
Get Next Interchange	<u>IviScope Get Next Interchange Warning</u>	IVI

Warning

Reset Interchange Check	IviScope Reset Interchange Check	IVI
-------------------------------	------------------------------------------------------	-----

Clear Interchange Warnings	IviScope Clear Interchange Warnings	IVI
-------------------------------	-------------------------------------------------------------	-----

Coercion Info

Get Next Coercion Record	IviScope Get Next Coercion Record	IVI
-----------------------------	-------------------------------------------------------	-----

Close	IviScope Close	VPP
-------	--------------------------------	-----

Related Topics

[IviScope Properties](#)

[IVI Inherent Class Capabilities](#)

IviScope Properties



Note You use the LabVIEW Property Node to get and set properties.

IviScope Base Properties

IviScope Acquisition Start Time

IviScope Acquisition Type

IviScope Channel Enabled

IviScope Horizontal Minimum Number of Points

IviScope Horizontal Record Length

IviScope Horizontal Sample Rate

IviScope Horizontal Time Per Record

IviScope Input Impedance

IviScope Maximum Input Frequency

IviScope Probe Attenuation

IviScope Trigger Coupling

IviScope Trigger Holdoff

IviScope Trigger Level

IviScope Trigger Slope

IviScope Trigger Source

IviScope Trigger Type

IviScope Vertical Coupling

IviScope Vertical Offset

IviScope Vertical Range

IviScopeInterpolation

[IviScope Interpolation](#)

IviScopeTVTrigger Properties

[IviScope TV Trigger Event](#)

[IviScope TV Trigger Line Number \[TV\]](#)

[IviScope TV Trigger Polarity \[TV\]](#)

[IviScope TV Trigger Signal Format \[TV\]](#)

IviScopeRuntTrigger Properties

[IviScope Runt High Threshold \[RT\]](#)

[IviScope Runt Low Threshold \[RT\]](#)

[IviScope Runt Polarity \[RT\]](#)

IviScopeGlitchTrigger Properties

[IviScope Glitch Condition \[GT\]](#)

[IviScope Glitch Polarity \[GT\]](#)

[IviScope Glitch Width \[GT\]](#)

IviScopeWidthTrigger Properties

[IviScope Width Condition \[WT\]](#)

[IviScope Width High Threshold \[WT\]](#)

[IviScope Width Low Threshold \[WT\]](#)

[IviScope Width Polarity \[WT\]](#)

IviScopeAcLineTrigger Properties

[IviScope AC Line Trigger Slope](#)

IviScopeMeasWaveform Properties

[IviScope Measurement High Reference \[WM\]](#)

[IviScope Measurement Low Reference \[WM\]](#)

[IviScope Measurement Mid Reference \[WM\]](#)

IviScopeMinMaxWaveform Properties

[IviScope Number of Envelopes \[MM\]](#)

[IviScope Probe Sense Value](#)

IviScopeContinuousAcquisition

IviScope Initiate Continuous

IviScopeAverageAcquisition

IviScope Number of Averages

IviScopeSampleMode

[IviScope Sample Mode](#)

IviScopeTriggerModifier Properties

[IviScope Trigger Modifier](#)

IviScope Property Value Definitions

The following table defines the value for each property constant in the IviScope class driver.

IviScope Property Defined Values

Property	Value Name	Defined Value	Description
VERTICAL COUPLING	AC	0	The oscilloscope AC coupled signal.
	DC	1	The oscilloscope DC coupled signal.
	GND	2	The oscilloscope ground input signal.
PROBE ATTENUATION	Probe Sense On	-1	<p>Setting this property to IVISCOPE_VAL_PROB configures the oscilloscope the attenuation of the probe automatically. After you enable automatic probe sense, queries of this property return IVISCOPE_VAL_PROB. Use the IviScope Probe Attenuation property to obtain the current probe attenuation.</p> <p>If you set the oscilloscope probe attenuation automatically, the probe attenuation value changes any time. When the oscilloscope detects a new probe or other settings in the oscilloscope also change. The driver knows when these changes occur. Therefore, when you enable automatic probe sense, the driver disables caching of properties that depend on the probe. These properties include Vertical Range, IviScope Offset, and all the properties that configure trigger level:</p>

			Trigger Level property
	Runt	3	Configures the oscilloscope triggering. A runt trigger occurs when the trigger signal crosses the runt thresholds twice within the other runt threshold. The runt thresholds with Runt High Threshold and Runt Low Threshold properties specify the polarity of IviScope Runt Polarity .
	Glitch	4	4
	TV	5	Configures the oscilloscope on TV signals. The user triggers with the IviScope Signal Format , IviScope Line Number , and IviScope Polarity properties.
	Immediate	6	The oscilloscope does not trigger of any kind.
	AC Line	7	The oscilloscope waits for the AC line.
TRIGGER SLOPE	Positive	1	A positive (rising) edge through the trigger level on the oscilloscope.
	Negative	0	A negative (falling) edge through the trigger level on the oscilloscope.
TRIGGER SOURCE	External	"VAL_EXTERNAL"	The oscilloscope waits for the external trigger.
	PXI TRIG0 or VXI TTL0	"VAL_TTL0"	The oscilloscope waits for a trigger on the TLL0 line.
	PXI TRIG1 or VXI TTL1	"VAL_TTL1"	The oscilloscope waits for a trigger on the TLL1 line.

	PXI TRIG2 or VXI TTL2	"VAL_TTL2"	The oscilloscope waits a trigger on the TLL2
	PXI TRIG3 or VXI TTL3	"VAL_TTL3"	The oscilloscope waits a trigger on the TLL3
	PXI TRIG4 or VXI TTL4	"VAL_TTL4"	The oscilloscope waits a trigger on the TLL4
	PXI TRIG5 or VXI TTL5	"VAL_TTL5"	The oscilloscope waits a trigger on the TLL5
	PXI TRIG6 or VXI TTL6	"VAL_TTL6"	The oscilloscope waits a trigger on the TLL6
	PXI TRIG7 or VXI TTL7	"VAL_TTL7"	The oscilloscope waits a trigger on the TLL7
	ECL0	"VAL_ECL0"	The oscilloscope waits a trigger on the ECL0
	ECL1	"VAL_ECL1"	The oscilloscope waits a trigger on the ECL1
	PXI Star	"VAL_PXI_STAR"	The oscilloscope waits a trigger on the PXI Si
	RTSI0	"VAL_RTSI_0"	The oscilloscope waits a trigger on the RTSI (
	RTSI1	"VAL_RTSI_1"	The oscilloscope waits a trigger on the RTSI :
	RTSI2	"VAL_RTSI_2"	The oscilloscope waits a trigger on the RTSI :
	RTSI3	"VAL_RTSI_3"	The oscilloscope waits a trigger on the RTSI :
	RTSI4	"VAL_RTSI_4"	The oscilloscope waits a trigger on the RTSI ,

	RTSI5	"VAL_RTSI_5"	The oscilloscope waits a trigger on the RTSI 5.
	RTSI6	"VAL_RTSI_6"	The oscilloscope waits a trigger on the RTSI 6.
TRIGGER COUPLING	AC	0	The oscilloscope AC couple trigger signal.
	DC	1	The oscilloscope DC couple trigger signal.
	HF Reject	3	The oscilloscope rejects high frequencies from the trigger signal.
	LF Reject	4	The oscilloscope rejects low frequencies from the trigger signal.
	Noise Reject	5	The oscilloscope rejects noise from the trigger signal.
INTERPOLATION	No Interpolation	1	The oscilloscope does not interpolate between data points in the waveform driver sets every element of the waveform array for which the oscilloscope cannot read an IEEE-defined NaN value.
	sin(x)/x	2	The oscilloscope uses sinc calculation to interpolate between data points when it can not resolve a waveform record.
	Linear Interpolation	3	The oscilloscope uses linear approximation to interpolate between data points when it can not resolve a waveform record.
TV TRIGGER SIGNAL FORMAT	NTSC	1	Sets the TV trigger signal format to NTSC.
	PAL	2	Sets the TV trigger signal format to PAL.
	SECAM	3	Sets the TV trigger signal format to SECAM.

			SECAM.
TV TRIGGER EVENT	Field 1	1	Field 1 of the video signal
	Field 2	2	Field 2 of the video signal
	Any Field	3	Any field of the video signal
	Any Line	4	Any line of the video signal
	Line Number	5	
TV TRIGGER POLARITY	Positive	1	Positive video sync pulse
	Negative	1	Negative video sync pulse
RUNT POLARITY	Positive	1	The oscilloscope trigger on a positive runt occurs when a rising edge crosses the high runt threshold and does not cross the low runt threshold before it crosses the low runt threshold.
	Negative	2	The oscilloscope trigger on a negative runt occurs when a falling edge crosses the high runt threshold and does not cross the low runt threshold before it crosses the high runt threshold.
	Either	3	The oscilloscope trigger on either positive or negative runt.
GLITCH POLARITY	Positive	1	The oscilloscope trigger on a positive glitch.
	Negative	2	The oscilloscope trigger on a negative glitch.
	Either	3	The oscilloscope trigger on either positive or negative glitch.
GLITCH CONDITION	Less Than	1	The oscilloscope trigger on a glitch that is less than the specified width.
	Greater	2	The oscilloscope trigger on a glitch that is greater than the specified width.

	Than		than glitch.
WIDTH POLARITY	Positive	1	Configures the oscilloscope to trigger on positive pulses that meet the condition specified by the IviScope Width property.
	Negative	2	Configures the oscilloscope to trigger on negative pulses that meet the condition specified by the IviScope Width property.
WIDTH CONDITION	Within	1	Configures the oscilloscope to trigger on pulses that have a width greater than the high threshold and less than the low threshold specified by the IviScope Width High Threshold and IviScope Width Low Threshold properties.
	Outside	2	Configures the oscilloscope to trigger on pulses that have a width either greater than the high threshold or less than the low threshold specified by the IviScope Width High Threshold and IviScope Width Low Threshold properties.
AC LINE TRIGGER SLOPE	Positive	1	Configures the oscilloscope to trigger on positive slope zero crossings of the network supply voltage.
	Negative	2	Configures the oscilloscope to trigger on negative slope zero crossings of the network supply voltage.
	Either	3	Configures the oscilloscope to trigger on either positive or negative slope zero crossings of the network supply voltage.

ACQUISITION TYPE	Normal	0	Configures the oscilloscope to acquire one sample for each point in the waveform record. The oscilloscope uses real-time or equivalent sampling.
	Peak Detect	1	Sets the oscilloscope to acquire in peak detect acquisition mode. The oscilloscope oversamples the input signal and keeps the maximum values that occur at each position in the waveform record. The oscilloscope uses equivalent sampling.
	High Resolution	2	Configures the oscilloscope to acquire and oversample the input signal. The oscilloscope calculates the average value that corresponds to each position in the waveform record. The oscilloscope uses only equivalent sampling.
	Envelope	3	Sets the oscilloscope to acquire in envelope acquisition mode. The oscilloscope acquires multiple waveforms and keeps the minimum and maximum voltages it acquires for each position in the waveform record. The number of waveforms the oscilloscope acquires with the IviScope.Acquire Envelopes property. The oscilloscope can use real-time or equivalent sampling.
	Average	4	Configures the oscilloscope to acquire multiple waveforms and average the average value for each position in the waveform record. You can specify the number of waveforms the oscilloscope acquires with the IviScope.Number of Waveforms property.

			property. The oscilloscope time or equivalent time
TRIGGER MODIFIER	No Trigger Modifier	1	The oscilloscope waits until you specify occurs.
	Auto	2	The oscilloscope auto if the configured trigger within the oscilloscope period.
	Auto Level	3	The oscilloscope adjusts level if the trigger you occur.
SAMPLE MODE	Real Time	0	Indicates that the oscilloscope real-time sampling.
	Equivalent Time	1	Indicates that the oscilloscope equivalent time sampling.

Related Topics

[IviScope VI Tree](#)

[IviScope Properties](#)

IviScope Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviScope Errors	0xBFFA2001 to 0xBFFA2003
IviScope Warnings	0x3FFA2001 to 0xBFFA2004
IVI Specific Driver Errors	0xBFFA4000 to 0xBFFA5FFF
IVI Specific Driver Warnings	0x3FFA4000 to 0x3FFA5FFF
IVI Errors	BFFA0000 to BFFA1FFF
IVI Warnings	3FFA0000 to 3FFA1FFF
Common Instrument Driver Errors	BFFC0000 to BFFCFFFF
Common Instrument Driver Warnings	3FFC0000 to 3FFCFFFF
VISA Errors	BFFF0000 to BFFFFFFF
VISA Warnings	3FFF0000 to 3FFFFFFF

The IviScope class driver defines the following error codes in addition to the IVI defined error codes.

IviScope Error Codes

Error	Value
IVISCOPE_ERROR_CHANNEL_NOT_ENABLED	0xBFFA2
IVISCOPE_ERROR_UNABLE_TO_PERFORM_MEASUREMENT	0xBFFA2
IVISCOPE_ERROR_MAX_TIME_EXCEEDED	0xBFFA2
IVISCOPE_ERROR_INVALID_ACQ_TYPE	0xBFFA2

IviScope Warning Code

Warning	Value	Message
IVISCOPE_WARN_INVALID_WFM_ELEMENT	0x3FFA2001	One of the elements in the waveform array is invalid.

Related Topic

[IVI Status Codes](#)

IviScope VIs

Expand this book to view an alphabetized list of IviScope VIs.

IviScope Abort

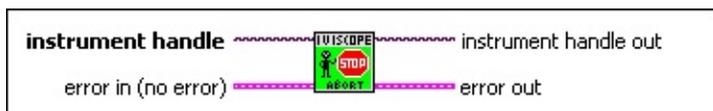
IviScope Base Capability Group

This VI aborts an acquisition and returns the oscilloscope to the Idle state. You initiate an acquisition with the [IviScope Initiate Acquisition](#) VI.



Notes

1. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query](#) VI at the conclusion of the sequence.
2. If the instrument cannot abort an initiated acquisition, this VI returns the IVI_ERROR_FUNCTION_NOT_SUPPORTED error.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Acquisition Status

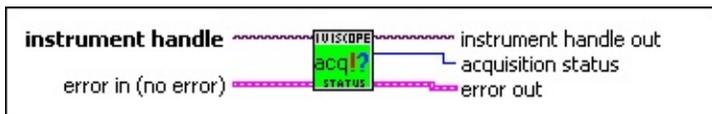
IviScope Base Capability Group

This VI returns whether an acquisition is in progress, complete, or if the status is unknown.



Notes

1. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query](#) VI at the conclusion of the sequence.
2. If the instrument cannot return its acquisition status, this VI returns the IVISCOPE_VAL_ACQ_STATUS_UNKNOWN value in the **Acquisition Status** parameter.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



acquisition status Returns the acquisition status of the oscilloscope. Possible values that this parameter returns are:

- `IVISCOPE_VAL_ACQ_IN_PROGRESS (0)`—The oscilloscope is currently acquiring a waveform.
- `IVISCOPE_VAL_ACQ_COMPLETE (1)`—The acquisition is complete.
- `IVISCOPE_VAL_ACQ_STATE_UNKNOWN (-1)`—The driver cannot determine the status of the acquisition.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Actual Record Length

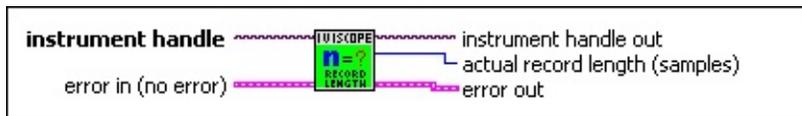
IviScope Base Capability Group

This VI returns the actual number of points the oscilloscope acquires for each channel. After you configure the oscilloscope for an acquisition, call this VI to determine the size of the waveforms that the oscilloscope acquires. The value is greater than or equal to the minimum number of points you specify in the [IviScope Configure Acquisition Record](#) VI.

You must allocate a ViReal64 array of this size or greater to pass as the **Waveform Array** parameter of the [IviScope Read Waveform](#) and [IviScope Fetch Waveform](#) VIs.



Note The oscilloscope may use records of different size depending on the acquisition type. You specify the acquisition type with the [IviScope Configure Acquisition Type](#) VI.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

actual record length Returns the actual number of points the oscilloscope acquires for each channel. The driver returns the value of the [IviScope Horizontal Record Length](#) property.

Err **error out** The error out cluster passes error or warning information

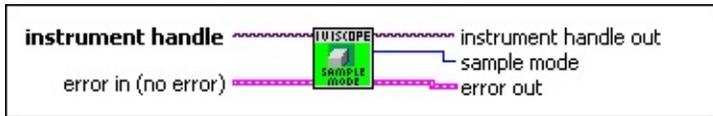
out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Actual Sample Mode

IviScopeSampleMode Capability Group

This VI returns the sample mode the oscilloscope is currently using.



Info **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

Error **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

Info **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

sample mode Returns the sample mode the oscilloscope is currently using. The driver returns the value of the [IviScope Sample Mode \[SM\]](#) property.

Defined Values:

IVISCOPE_VAL_REAL_TIME (0)—Indicates that the oscilloscope is using real-time sampling.

IVISCOPE_VAL_EQUIVALENT_TIME (1)—Indicates that the oscilloscope is using equivalent-time sampling.

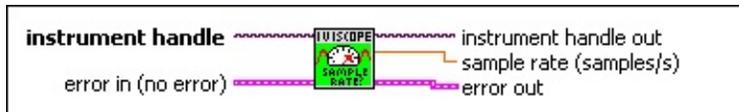
Error **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Actual Sample Rate

IviScope Base Capability Group

This VI returns the effective sample rate of the acquired waveform using the current configuration in samples per second.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

sample rate Returns the effective sample rate of the acquired waveform the oscilloscope acquires for each channel. The driver returns the value of the [IviScope Horizontal Sample Rate](#) property.

Units: samples per second

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

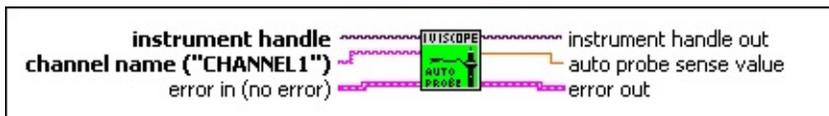
IviScope Auto Probe Sense Value

IviScopeProbeAutoSense Capability Group

The VI returns the probe attenuation value the oscilloscope senses. You enable the automatic probe sense capability by setting the **Probe Attenuation** parameter of the IviScope Configure Vertical VI to IVISCOPE_VAL_PROBE_SENSE_ON.



Note If you disable the automatic probe sense capability, this VI returns the manual probe attenuation setting.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



channel name ("CHANNEL1") Pass the virtual channel name that you assign to the instrument in MAX for which you want to query the probe sense value.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



auto probe sense value Returns the probe attenuation value the oscilloscope senses. The driver returns the value of the [IviScope Probe Sense Value](#) property.



Note If you disable the automatic probe sense capability, this parameter returns the manual probe attenuation setting.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Auto Setup

IviScopeAutoSetup Capability Group [AS]

This VI automatically configures the instrument.



Note When you call this VI, the oscilloscope senses the input signal and automatically configures many of the instrument settings. The settings no longer match the cache values for the corresponding properties. Therefore, this VI invalidates all property cache values.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

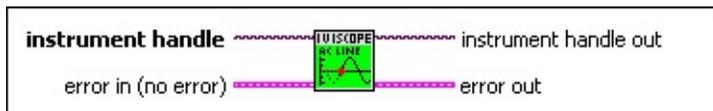
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure AC Line Trigger Slope

IviScopeAcLineTrigger Capability Group [AT]

This VI configures the slope of the AC Line trigger.

 **Note** This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_AC_LINE_TRIGGER`. Call the [IviScope Configure Trigger](#) VI to set the trigger type before calling this VI.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

trigger slope Specify whether you want the oscilloscope to trigger on a zero crossing with a positive, negative, or either slope of the network supply voltage. The driver uses this value to set the [IviScope AC Line Trigger Slope \[AT\]](#) property.

Defined Values:

`IVISCOPE_VAL_AC_LINE_POSITIVE`—Positive crossing

`IVISCOPE_VAL_AC_LINE_NEGATIVE`—Negative crossing

`IVISCOPE_VAL_AC_LINE_EITHER`—Either crossing

Default value: `IVISCOPE_VAL_AC_LINE_POSITIVE`

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#)

VI.



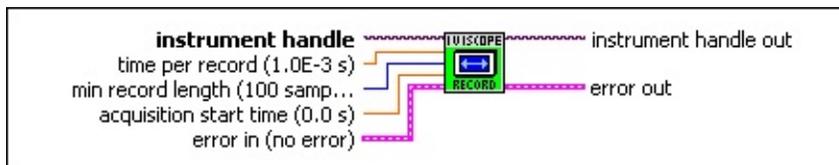
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Acquisition Record

IviScope Base Capability Group

This VI configures the most commonly configured properties of the oscilloscope acquisition subsystem. These properties are the time per record, minimum record length, and the acquisition start time.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

time per record (seconds) Pass the time per record. The driver uses this value to set the [IviScope Horizontal Time Per Record](#) property.

Units: seconds

minimum record length Pass the minimum number of points you allow for in the waveform recorded. The driver uses this value to set the [IviScope Horizontal Minimum Number of Points](#) property. Call the [IviScope Actual Record Length](#) VI to obtain the actual record length.



Note The oscilloscope may use records of different size depending on the acquisition type. You specify the acquisition type with the [IviScope Configure Acquisition Type](#) VI.

start time (seconds) Pass the position of the first point in the waveform record relative to the trigger event. The driver uses this value to set the [IviScope Acquisition Start Time](#) property.

Specify the length of time from the trigger event to the first point in the waveform record. If this value is positive, the first point in the waveform record occurs after the trigger event. If this value is negative, the first point in the waveform record occurs before the

trigger event.

Units: seconds

Default: 0.0



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



[instrument handle out](#) The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



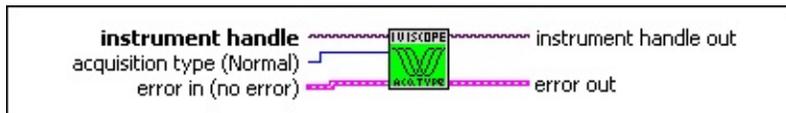
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Acquisition Type

IviScope Base Capability Group

This VI configures how the oscilloscope acquires data and how it fills the waveform record.



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- acquisition type** Specify the manner in which you want the oscilloscope to acquire data and fill the waveform record. The driver sets the [IviScope Acquisition Type](#) property to this value.

Defined Values:

IVISCOPE_VAL_NORMAL—Sets the oscilloscope to the normal acquisition mode. The oscilloscope acquires one sample for each point in the waveform record. The oscilloscope can use real-time or equivalent-time sampling.

IVISCOPE_VAL_PEAK_DETECT—Sets the oscilloscope to the peak-detect acquisition mode. The oscilloscope oversamples the input signal and keeps the minimum and maximum values that correspond to each position in the waveform record. The oscilloscope uses only real-time sampling.

IVISCOPE_VAL_HI_RES—Sets the oscilloscope to the high-resolution acquisition mode. The oscilloscope oversamples the input signal and calculates an average value for each position in the waveform record. The oscilloscope uses only real-time sampling.

IVISCOPE_VAL_ENVELOPE—Sets the oscilloscope to the envelope acquisition mode. The oscilloscope acquires multiple waveforms and keeps the minimum and maximum voltages it acquires for each point in the waveform record. You specify the

number of waveforms the oscilloscope acquires with the [IviScope Configure Number of Envelopes \[MmW\]](#) VI. The oscilloscope can use real-time or equivalent-time sampling.

IVISCOPE_VAL_AVERAGE—Sets the oscilloscope to the average acquisition mode. The oscilloscope acquires multiple waveforms and calculates an average value for each point in the waveform record. You specify the number of waveforms the oscilloscope acquires with the [IviScope Configure Number of Averages \[AA\]](#) VI. The oscilloscope can use real-time or equivalent-time sampling.

Default value: IVISCOPE_VAL_NORMAL



Note When you set this parameter to IVISCOPE_VAL_ENVELOPE or IVISCOPE_VAL_PEAK_DETECT, the oscilloscope acquires minimum and maximum waveforms. To retrieve the minimum and maximum waveforms, use the [IviScope Read Min Max Waveform \[MmW\]](#) and [IviScope Fetch Min Max Waveform \[MmW\]](#) VIs.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



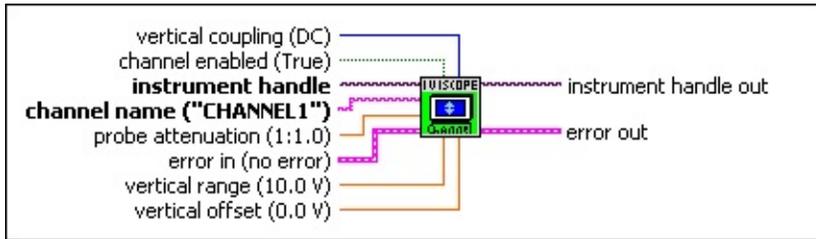
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Channel

IviScope Base Capability Group

This VI configures the most commonly configured properties of the oscilloscope channel sub-system. These properties are the range, offset, coupling, probe attenuation, and whether the channel is enabled.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

abc **channel name** Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

vertical range Pass the value of the input range the oscilloscope uses for the channel. The driver sets the [IviScope Vertical Range](#) property to this value.

For example, to acquire a sine wave that spans between -5.0 and 5.0 V, pass 10.0 as the value of this parameter.

Units: volts

vertical offset Pass the location of the center of the range that you

specify with the **Vertical Range** parameter. Express the value with respect to ground. The driver sets the IviScope [Vertical Offset](#) to this value.

For example, to acquire a sine wave that spans between 0.0 and 10.0 V, pass 5.0 as the value of this parameter.

Units: volts

Default value: 0.0

- vertical coupling** Specify how you want the oscilloscope to couple the input signal for the channel. The driver sets the [IviScope Vertical Coupling](#) property to this value.

Defined Values:

IVISCOPE_VAL_AC—AC Coupling

IVISCOPE_VAL_DC—DC Coupling

IVISCOPE_VAL_GND—Ground Coupling

Default value: IVISCOPE_VAL_DC

- probe attenuation** Pass the scaling factor by which the probe you attach to the channel attenuates the input. For example, when you use a 10:1 probe, set this parameter to 10.0. The driver sets the [IviScope Probe Attenuation](#) property to this value.

This driver reserves negative values to control the oscilloscope's automatic probe sense capability. Setting this parameter to IVISCOPE_VAL_PROBE_SENSE_ON configures the oscilloscope to sense the attenuation of the probe automatically.

Defined Values:

IVISCOPE_VAL_PROBE_SENSE_ONE—enables the oscilloscope's automatic probe sense capability.

Default Value: 1.0



Notes

1. If the oscilloscope is set to sense the probe attenuation automatically, setting this parameter to a positive value disables the automatic probe sense and configures the oscilloscope to use the manual probe attenuation you specify.

2. If you use a manual probe attenuation, you must set the probe attenuation to reflect the new probe attenuation each time you attach a different probe.
3. Use the [IviScope Auto Probe Sense Value \[PAS\]](#) VI to obtain the actual probe attenuation while the automatic probe sense capability is enabled.
4. If you set the oscilloscope to sense the probe attenuation automatically, the probe attenuation value can change at any time. When the oscilloscope detects a new probe attenuation value, other settings in the oscilloscope might also change. The driver has no way of knowing when these changes occur. Therefore, when you enable the automatic probe sense capability, this driver disables caching for properties that depend on the probe attenuation. These properties include [IviScope Vertical Range](#), [IviScope Vertical Offset](#), and all the properties that configure trigger levels. To maximize performance, set this parameter to a manual probe attenuation setting.

channel enabled Specify whether you want the oscilloscope to acquire a waveform for the channel when you call the [IviScope Initiate Acquisition](#), [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), or [IviScope Read Waveform Measurement \[WM\]](#) VI. The driver sets the [IviScope Channel Enabled](#) property to this value.

Default Value: TRUE

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

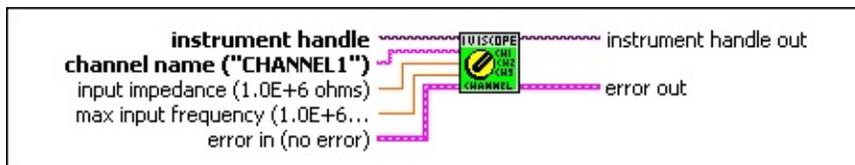
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Channel Characteristics

IviScope Base Capability Group

This VI configures the properties that control the electrical characteristics of the channel. These properties are the input impedance and the maximum frequency of the input signal.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

abc **channel name** Pass the virtual channel name that you assign to the instrument in MAX.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

input impedance (ohms) Pass the input impedance you want to use for the channel. The driver sets the [IviScope Input Impedance](#) to this value.

Units: ohms

Default Value: 1000000.0

max input frequency (hertz) Pass the maximum frequency for the input signal you want the instrument to accommodate without

attenuating it by more than 3dB. The driver sets the [IviScope Maximum Input Frequency](#) to this value.

Units: hertz (Hz)

 **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

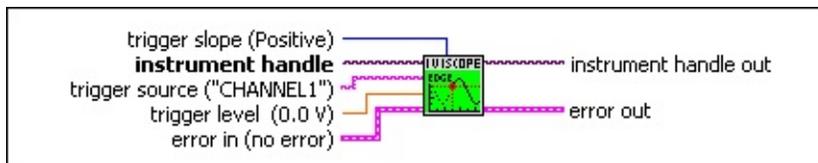
 **[error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Edge Trigger Source

IviScope Base Capability Group

This VI configures the edge trigger. An edge trigger occurs when the trigger signal passes through the voltage threshold that you specify with the **Trigger Level** parameter and has the slope that you specify with the **Trigger Slope** parameter.



Notes

1. This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_EDGE_TRIGGER`. Call the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs to set the trigger type and trigger coupling before calling this VI.
2. If the trigger source is one of the analog input channels, you must configure the vertical range, vertical offset, vertical coupling, probe attenuation, and the maximum input frequency before calling this VI.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



trigger source Pass the source you want the oscilloscope to monitor for a trigger. The driver sets the [IviScope Trigger Source](#) property to this value.

After you call one of the [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), [IviScope Read Waveform Measurement \[WM\]](#), or [IviScope Initiate Acquisition](#) VIs, the oscilloscope waits for the trigger from the source you specify in this parameter.

This control accepts one of the valid channel names or additional trigger sources listed below.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Additional Trigger Sources

IVISCOPE_VAL_EXTERNALExternal

IVISCOPE_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVISCOPE_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVISCOPE_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVISCOPE_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVISCOPE_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVISCOPE_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVISCOPE_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVISCOPE_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVISCOPE_VAL_ECL0—VXI ECL0

IVISCOPE_VAL_ECL1—VXI ECL1

IVISCOPE_VAL_PXI_STAR—PXI Star

IVISCOPE_VAL_RTSI_0—RTSI line 0

IVISCOPE_VAL_RTSI_1—RTSI line 1

IVISCOPE_VAL_RTSI_2—RTSI line 2

IVISCOPE_VAL_RTSI_3—RTSI line 3

IVISCOPE_VAL_RTSI_4—RTSI line 4

IVISCOPE_VAL_RTSI_5—RTSI line 5

IVISCOPE_VAL_RTISI_6—RTSI line 6



Note IVISCOPE_VAL_EXTERNAL—The oscilloscope waits for a trigger on the external trigger input.



trigger level (volts) Pass the voltage threshold you want the oscilloscope to use for edge triggering. The driver sets the [IviScope Trigger Level](#) property to this value.

The oscilloscope triggers when the trigger signal passes through the threshold you specify with this parameter and has the slope you specify with the **Trigger Slope** parameter.

Units: volts

Default value: 0.0



Note This parameter affects instrument behavior only when you select a channel or the external trigger input as the Trigger Source. You cannot configure the trigger level that the oscilloscope uses for other trigger sources. For example, if you select IVISCOPE_VAL_AC_LINE_TRIGGER as the trigger source, the oscilloscope triggers on zero crossings.



trigger slope Specify whether you want a rising edge or a falling edge passing through the Trigger Level to trigger the oscilloscope. The driver sets the [IviScope Trigger Slope](#) property to this value.

Defined Values:

IVISCOPE_VAL_POSITIVE Rising edge

IVISCOPE_VAL_NEGATIVE Falling edge

Default value: IVISCOPE_VAL_POSITIVE



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

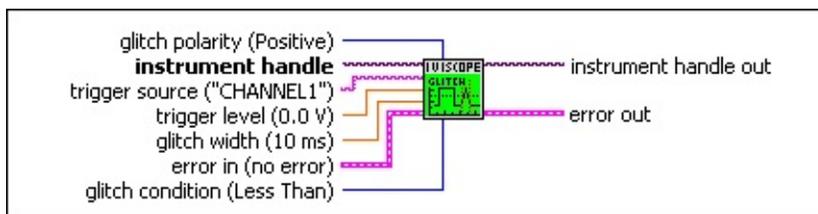
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Glitch Trigger Source

IviScopeGlitchTrigger Capability Group [GT]

This VI configures the glitch trigger. A glitch trigger occurs when the trigger signal has a pulse with a width that is less than the glitch width. You specify the glitch width in the **Glitch Width** parameter. You specify the polarity of the pulse with the **Glitch Polarity** parameter. The trigger does not actually occur until the edge of a pulse that corresponds to the Glitch Width and Glitch Polarity crosses the threshold you specify in the **Trigger Level** parameter.

 **Note** This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_GLITCH_TRIGGER`. Call the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs to set the trigger type and trigger coupling before calling this VI.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

abc **trigger source** Pass the source you want the oscilloscope to monitor for a trigger. The driver sets the [IviScope Trigger Source](#) property to this value.

After you call one of the [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), [IviScope Read Waveform Measurement \[WM\]](#), or [IviScope Initiate Acquisition](#) VIs, the oscilloscope waits for the trigger from the source you specify in this parameter.

This control accepts one of the valid channel names or additional trigger sources listed below.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Additional Trigger Sources

IVISCOPE_VAL_EXTERNAL—External
IVISCOPE_VAL_TTL0—PXI TRIG0 or VXI TTL0
IVISCOPE_VAL_TTL1—PXI TRIG1 or VXI TTL1
IVISCOPE_VAL_TTL2—PXI TRIG2 or VXI TTL2
IVISCOPE_VAL_TTL3—PXI TRIG3 or VXI TTL3
IVISCOPE_VAL_TTL4—PXI TRIG4 or VXI TTL4
IVISCOPE_VAL_TTL5—PXI TRIG5 or VXI TTL5
IVISCOPE_VAL_TTL6—PXI TRIG6 or VXI TTL6
IVISCOPE_VAL_TTL7—PXI TRIG7 or VXI TTL7
IVISCOPE_VAL_ECL0—VXI ECL0
IVISCOPE_VAL_ECL1—VXI ECL1
IVISCOPE_VAL_PXI_STAR—PXI Star
IVISCOPE_VAL_RTSI_0—RTSI line 0
IVISCOPE_VAL_RTSI_1—RTSI line 1
IVISCOPE_VAL_RTSI_2—RTSI line 2
IVISCOPE_VAL_RTSI_3—RTSI line 3
IVISCOPE_VAL_RTSI_4—RTSI line 4
IVISCOPE_VAL_RTSI_5—RTSI line 5
IVISCOPE_VAL_RTSI_6—RTSI line 6



Note IVISCOPE_VAL_EXTERNAL—The oscilloscope waits for a

trigger on the external trigger input.

- trigger level (volts)** Pass the voltage threshold you want the oscilloscope to use for glitch triggering. The driver sets the [IviScope Trigger Level](#) property to this value.

The oscilloscope triggers when a glitch crosses the trigger threshold you specify with this parameter.

Units: volts

Default value: 0.0



Note This parameter only affects instrument behavior when you select a channel or the external trigger input as the Trigger Source. You cannot configure the trigger level that the oscilloscope uses for other trigger sources.

- glitch width (seconds)** Pass the length of time you want the oscilloscope to use for the glitch width. The driver sets the [IviScope Glitch Width \[GT\]](#) property to this value.

The oscilloscope triggers when it detects a pulse with a width less than or greater than this value, depending on the **Glitch Condition** parameter.

Units: seconds

Default value: 0.000010

- glitch polarity** Pass the polarity of the glitch that you want to trigger the oscilloscope. The driver sets the [IviScope Glitch Polarity \[GT\]](#) property to this value.

Defined Values:

IVISCOPE_VAL_GLITCH_POSITIVE—The oscilloscope triggers on a positive glitch.

IVISCOPE_VAL_GLITCH_NEGATIVE—The oscilloscope triggers on a negative glitch.

IVISCOPE_VAL_GLITCH_EITHER—The oscilloscope triggers on either a positive or negative glitch.

Default Value: IVISCOPE_VAL_GLITCH_POSITIVE

- glitch condition** Pass the glitch condition. The driver uses this value to set the [IviScope Glitch Condition \[GT\]](#) property.

The oscilloscope triggers when it detects a pulse with a width less than or greater than this value, depending on this parameter.

Defined Values:

IVISCOPE_VAL_GLITCH_LESS_THAN—The oscilloscope triggers when the pulse width is less than the Glitch Width.

IVISCOPE_VAL_GLITCH_GREATER_THAN—The oscilloscope triggers when the pulse width is greater than the Glitch Width.

Default Value: IVISCOPE_VAL_GLITCH_LESS_THAN

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

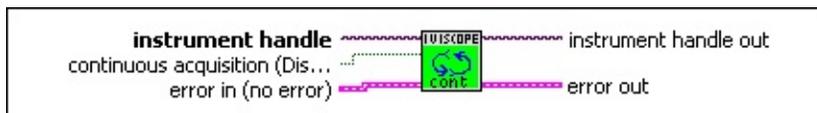
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Initiate Continuous

IviScopeContinuousAcquisition Capability Group [CA]

This VI configures the oscilloscope to perform continuous acquisition.



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- continuous acquisition** Specifies whether you want to enable continuous acquisition on the oscilloscope. The driver uses this value to set the [IviScope Initiate Continuous \[CA\]](#) property.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

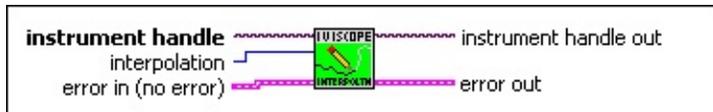
- instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Interpolation

IviScopeInterpolation Capability Group [I]

This VI configures the interpolation method the oscilloscope uses when it cannot sample a voltage for a point in the waveform record.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options \[I\]](#) VI.

interpolation Pass the interpolation method you want the oscilloscope to

use when it cannot sample a voltage for a point in the waveform record. The driver sets the [IviScope Interpolation](#) property to this value.

Defined Values:

IVISCOPE_VAL_NO_INTERPOLATION No interpolation

IVISCOPE_VAL_SINE_X $\sin(x)/x$ interpolation

IVISCOPE_VAL_LINEAR Linear interpolation

Default value: IVISCOPE_VAL_SINE_X

 **Note** IVISCOPE_VAL_NO_INTERPOLATION The oscilloscope does not interpolate any points in the waveform.

Instead, the driver sets every element in the waveform array for which the oscilloscope cannot sample a value to an IEEE-defined NaN (Not a Number) value.

[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

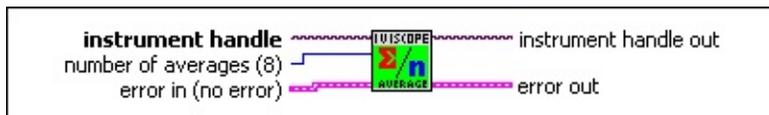
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Number of Averages

IviScopeAverageAcquisition Capability Group [AA]

When you set the acquisition type to `IVISCOPE_VAL_AVERAGE`, the oscilloscope acquires multiple waveforms. After each waveform acquisition, the oscilloscope keeps the average value of all acquisitions for each element in the waveform record. This VI configures the number of waveforms that the oscilloscope acquires and averages. After the oscilloscope acquires as many waveforms as you specify, it returns to the Idle state.

 **Note** Set the acquisition type to `IVISCOPE_VAL_AVERAGE` before you call this VI. To set the acquisition type, call the [IviScope Configure Acquisition Type](#) VI.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

number of averages When you set the acquisition type to `IVISCOPE_VAL_AVERAGE`, the oscilloscope acquires multiple waveforms. After each waveform acquisition, the oscilloscope keeps the average value of all acquisitions for each element in the waveform record. This parameter specifies the number of waveforms the oscilloscope acquires and averages. The driver sets the [IviScope Number of Averages \[AA\]](#) property to this value.

Default value: 8

 **Note** This parameter affects instrument behavior only when the acquisition type is set to `IVISCOPE_VAL_AVERAGE`.

error in (no error) The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

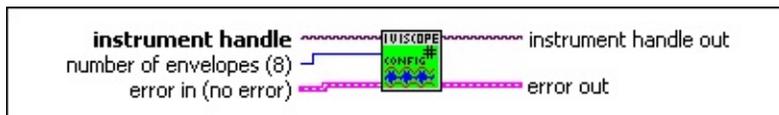
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Number of Envelopes

IviScopeMinMaxWaveform Capability Group [MmW]

When you set the acquisition type to `IVISCOPE_VAL_ENVELOPE`, the oscilloscope acquires multiple waveforms. After each waveform acquisition, the oscilloscope keeps the minimum and maximum values it finds for each element in the waveform record. This VI configures the number of waveforms the oscilloscope acquires and analyzes to create the minimum and maximum waveforms. After the oscilloscope acquires as many waveforms as you specify, it returns to the Idle state.

 **Note** Set the acquisition type to `IVISCOPE_VAL_ENVELOPE` before you call this VI. To set the acquisition type, call the [IviScope Configure Acquisition Type](#) VI.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

number of envelopes When you set the acquisition type to `IVISCOPE_VAL_ENVELOPE`, the oscilloscope acquires multiple waveforms. After each waveform acquisition, the oscilloscope keeps the minimum and maximum values it finds for each point in the waveform record. This parameter specifies the number of waveforms the oscilloscope acquires and analyzes to create the minimum and maximum waveforms. The driver sets the [IviScope Number of Envelopes \[MmW\]](#) property to this value.

Default value: 8

 **Note** This parameter affects instrument behavior only when the acquisition type is set to `IVISCOPE_VAL_ENVELOPE`.

- [error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

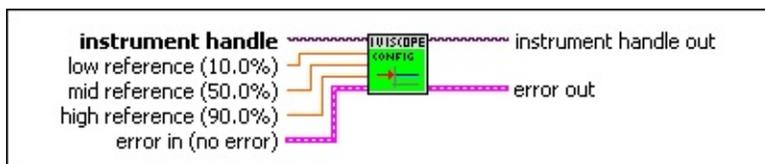
- [error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Reference Levels

IviScopeWaveformMeas Capability Group [WM]

This VI configures the reference levels for waveform measurements. You must call this VI before you call the [IviScope Read Waveform Measurement](#) VI to take measurements.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

low ref (percentage) Pass the low reference you want the oscilloscope to use for waveform measurements. The driver sets the [IviScope Measurement Low Reference \[WM\]](#) property to this value.

Units: A percentage of the difference between the Voltage High and Voltage Low.

 **Note** The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.

mid ref (percentage) Pass the mid reference you want the oscilloscope to use for waveform measurements. The driver sets the [IviScope Measurement Mid Reference \[WM\]](#) property to this value.

Units: A percentage of the difference between the Voltage High and Voltage Low.



Note The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.



high ref (percentage) Pass the high reference you want the oscilloscope to use for waveform measurements. The driver sets the [IviScope Measurement High Reference \[WM\]](#) property to this value.

Units: A percentage of the difference between the Voltage High and Voltage Low.



Note The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

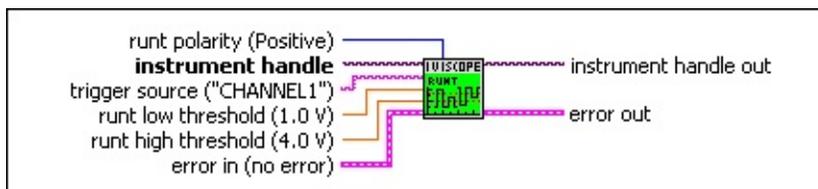
IviScope Configure Runt Trigger Source

IviScopeRuntTrigger Capability Group [RT]

This VI configures the runt trigger. A runt trigger occurs when the trigger signal crosses one of the runt thresholds twice without crossing the other runt threshold. You specify the runt thresholds with the **Runt Low Threshold** and **Runt High Threshold** parameters. You specify the polarity of the runt with the **Runt Polarity** parameter.



Note This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_RUNT_TRIGGER`. Call the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs to set the trigger type and trigger coupling before calling this VI.



I/O **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

abc **trigger source** Pass the source you want the oscilloscope to monitor for a trigger. The driver sets the [IviScope Trigger Source](#) property to this value.

After you call one of the [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), [IviScope Read Waveform Measurement \[WM\]](#), or [IviScope Initiate Acquisition](#) VIs, the oscilloscope waits for the trigger from the source you specify in this parameter.

This control accepts one of the valid channel names or additional trigger sources listed below.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one

instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Additional Trigger Sources

IVISCOPE_VAL_EXTERNALExternal

IVISCOPE_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVISCOPE_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVISCOPE_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVISCOPE_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVISCOPE_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVISCOPE_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVISCOPE_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVISCOPE_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVISCOPE_VAL_ECL—VXI ECL0

IVISCOPE_VAL_ECL1—VXI ECL1

IVISCOPE_VAL_PXI_STAR—PXI Star

IVISCOPE_VAL_RTISI_0—RTSI line 0

IVISCOPE_VAL_RTISI_1—RTSI line 1

IVISCOPE_VAL_RTISI_2—RTSI line 2

IVISCOPE_VAL_RTISI_3—RTSI line 3

IVISCOPE_VAL_RTISI_4—RTSI line 4

IVISCOPE_VAL_RTISI_5—RTSI line 5

IVISCOPE_VAL_RTISI_6—RTSI line 6



Note IVISCOPE_VAL_EXTERNALTThe oscilloscope waits for a trigger on the external trigger input.



runt low threshold (volts) Pass the low threshold you want the oscilloscope to use for runt triggering. The driver sets the [IviScope](#)

[Runt Low Threshold \[RT\]](#) property to this value.

Units: volts

Default Value: 1.0

- runt high threshold (volts)** Pass the high threshold you want the oscilloscope to use for runt triggering. The driver sets the [IviScope Runt High Threshold \[RT\]](#) property to this value.

Units: volts

Default Value: 4.0

- runt polarity** Pass the polarity of the runt that you want to trigger the oscilloscope. The driver sets the [IviScope Runt Polarity \[RT\]](#) property to this value.

Defined Values:

- `IVISCOPE_VAL_RUNT_POSITIVE`The oscilloscope triggers on a positive runt. A positive runt occurs when a rising edge crosses the Runt Low Threshold and does not cross the Runt High Threshold before re-crossing the Low Runt Threshold.
- `IVISCOPE_VAL_RUNT_NEGATIVE`The oscilloscope triggers on a negative runt. A negative runt occurs when a falling edge crosses the Runt High Threshold and does not cross the Runt Low Threshold before re-crossing the High Runt Threshold.
- `IVISCOPE_VAL_RUNT_EITHER`The oscilloscope triggers on either a positive or negative runt.

Default Value: `IVISCOPE_VAL_RUNT_POSITIVE`

-  **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



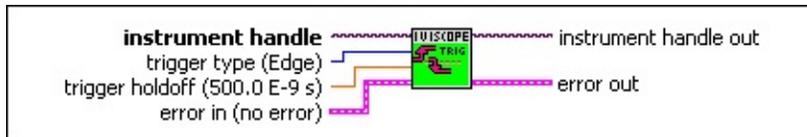
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Trigger

IviScope Base Capability Group

This VI configures the common properties of the trigger subsystem. These properties are the trigger type and holdoff. [Details](#)



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- trigger type** Pass the type of trigger you want the oscilloscope to use. The driver sets the [IviScope Trigger Type](#) to this value.

Defined Values:

IVISCOPE_VAL_EDGE_TRIGGER—Configures the oscilloscope for edge triggering. An edge trigger occurs when the trigger signal crosses the trigger level you specify with the slope you specify. You configure the trigger level and slope with the [IviScope Configure Edge Trigger Source](#) VI.

IVISCOPE_VAL_TV_TRIGGER—Configures the oscilloscope for TV triggering. You configure the TV signal type, the event on which to trigger, and the signal polarity with the [IviScope Configure TV Trigger Source \[TV\]](#) and [IviScope Configure TV Trigger Line Number \[TV\]](#) VIs.

IVISCOPE_VAL_RUNT_TRIGGER—Configures the oscilloscope for runt triggering. A runt trigger occurs when the trigger signal crosses one of the runt thresholds twice without crossing the other runt threshold. You configure the runt thresholds and the polarity of the runt with the [IviScope Configure Runt Trigger Source \[RT\]](#) VI.

IVISCOPE_VAL_GLITCH_TRIGGER—Configures the oscilloscope for glitch triggering. A glitch trigger occurs when the trigger signal has a pulse with a width that is less than the glitch width. The trigger does not actually occur until the edge of the pulse that corresponds to the glitch width and polarity you specify crosses the

trigger level. You configure the glitch width, the polarity of the pulse, and the trigger level with the [IviScope Configure Glitch Trigger Source \[GT\]](#) VI.

IVISCOPE_VAL_WIDTH_TRIGGER—Configures the oscilloscope for width triggering. A width trigger occurs when the oscilloscope detects a positive or negative pulse with a width between, or optionally outside, the width thresholds. The trigger does not actually occur until the edge of a pulse that corresponds to the width thresholds and polarity you specify crosses the trigger level. You configure the width thresholds, whether to trigger on pulse widths that are within or outside the width thresholds, the polarity of the pulse, and the trigger level with the [IviScope Configure Width Trigger Source \[WT\]](#) VI.

IVISCOPE_VAL_AC_LINE_TRIGGER—Configures the oscilloscope for AC Line triggering. You configure the slope on which to trigger with the [IviScope Configure AC Line Trigger Slope \[AT\]](#) VI.

IVISCOPE_VAL_IMMEDIATE_TRIGGER—Configures the oscilloscope for immediate triggering. The oscilloscope does not wait for trigger of any kind upon initialization.

Default Value: IVISCOPE_VAL_EDGE_TRIGGER



Note This control configures the type of trigger the oscilloscope uses. After you call this VI, you must call the appropriate trigger configuration VI to completely specify the trigger.

- trigger holdoff (seconds)** Pass the length of time you want the oscilloscope to wait after it detects a trigger until the oscilloscope enables the trigger subsystem to detect another trigger. The driver sets the [IviScope Trigger Holdoff](#) property to this value.

This parameter affects instrument operation only when the oscilloscope requires multiple acquisitions to build a complete waveform. The oscilloscope requires multiple waveform acquisitions when it uses equivalent-time sampling or when you set the acquisition type to envelope or average.

Units: seconds

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Trigger Details

When you call [IviScope Read Waveform](#), [IviScope Read Waveform Measurement \[WM\]](#), [IviScope Read Min Max Waveform \[MmW\]](#), or [IviScope Initiate Acquisition](#), the oscilloscope waits for a trigger. You specify the type of trigger for which the oscilloscope waits with the **Trigger Type** parameter.

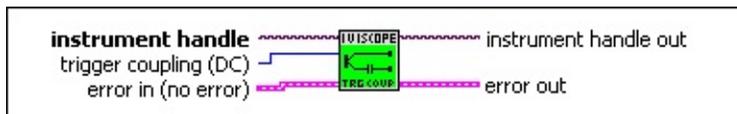
If the oscilloscope requires multiple waveform acquisitions to build a complete waveform, it waits for the length of time you specify with the **Holdoff** parameter to elapse since the previous trigger. The oscilloscope then waits for the next trigger. Once the oscilloscope acquires a complete waveform, it returns to the Idle state.

 **Note** After you call this VI, you must call the trigger configuration VI that corresponds to the Trigger Type you select to completely specify the trigger. For example, if you set the Trigger Type to `IVISCOPE_VAL_EDGE_TRIGGER`, you use the [IviScope Configure Edge Trigger Source](#) VI to completely specify the trigger.

IviScope Configure Trigger Coupling

IviScope Base Capability Group

This VI configures the trigger coupling.



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- trigger coupling** Pass the trigger coupling. The driver uses this value to set the [IviScope Trigger Coupling](#) property.

Defined Values:

IVISCOPE_VAL_AC—The oscilloscope AC couples the trigger signal.

IVISCOPE_VAL_DC—The oscilloscope DC couples the trigger signal.

IVISCOPE_VAL_LF_REJECT—The oscilloscope filters out the low frequencies from the trigger signal.

IVISCOPE_VAL_HF_REJECT—The oscilloscope filters out the high frequencies from the trigger signal.

IVISCOPE_VAL_NOISE_REJECT—The oscilloscope filters out the noise from the trigger signal.

Default Value: IVISCOPE_VAL_DC

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



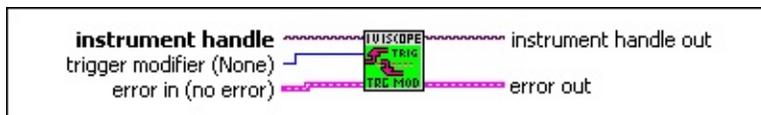
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Trigger Modifier

IviScopeTriggerModifier Capability Group [TM]

This VI configures the trigger modifier. The trigger modifier determines the oscilloscope's behavior in the absence of the configured trigger.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

trigger modifier Pass the trigger modifier. The driver uses this value to set the [IviScope Trigger Modifier \[TM\]](#) property.

Defined Values:

IVISCOPE_VAL_NO_TRIGGER_MOD—The oscilloscope waits until the trigger you specify occurs.

IVISCOPE_VAL_AUTO—The oscilloscope automatically triggers if the trigger you specify does not occur within the oscilloscope's timeout period.

IVISCOPE_VAL_AUTO_LEVEL—The oscilloscope adjusts the trigger level if the trigger you specify does not occur.

Default Value: IVISCOPE_VAL_NO_TRIGGER_MOD

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#)

VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

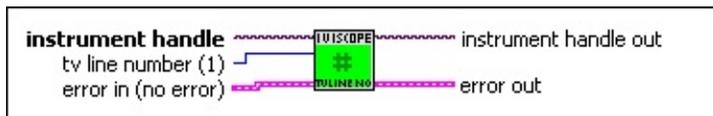
IviScope Configure TV Trigger Line Number

IviScopeTVTrigger Capability Group [TV]

This VI configures the TV line upon which the oscilloscope triggers. The line number is absolute and not relative to the field of the TV signal.



Note This VI affects instrument behavior only if the trigger type is set to `IVISCOPE_VAL_TV_TRIGGER` and the TV trigger event is set to `IVISCOPE_VAL_TV_LINE_NUMBER`. Call the [IviScope Configure Trigger](#) and the [IviScope Configure TV Trigger Source \[TV\]](#) VIs to set the trigger type and TV trigger event before calling this VI.



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



line number Pass the line on which you want the oscilloscope to trigger. The driver sets the [IviScope TV Trigger Line Number \[TV\]](#) property to this value.

The line number is independent of the field. This means that to trigger on the first line of the second field, you must specify a line number of 263 (if we assume that field one has 262 lines).

Default value: 1



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain

from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

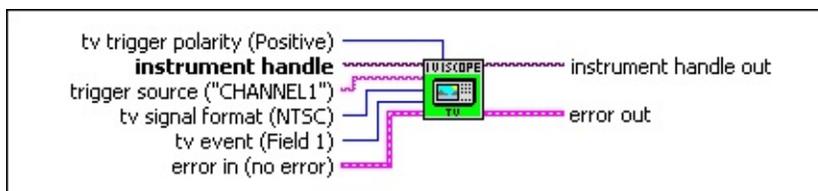
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure TV Trigger Source

IviScopeTVTrigger Capability Group [TV]

This VI configures the oscilloscope for TV triggering. It configures the TV signal format, the event, and the signal polarity.

-  **Note** This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_TV_TRIGGER`. Call the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs to set the trigger type and trigger coupling before calling this VI.



-  **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
-  **trigger source** Pass the source you want the oscilloscope to monitor for a trigger. The driver sets the [IviScope Trigger Source](#) property to this value.

After you call one of the [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), [IviScope Read Waveform Measurement \[WM\]](#), or [IviScope Initiate Acquisition](#) VIs, the oscilloscope waits for the trigger from the source you specify in this parameter.

This control accepts one of the valid channel names or additional trigger sources listed below.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an

instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Additional Trigger Sources

IVISCOPE_VAL_EXTERNAL—External

IVISCOPE_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVISCOPE_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVISCOPE_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVISCOPE_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVISCOPE_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVISCOPE_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVISCOPE_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVISCOPE_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVISCOPE_VAL_ECL0—VXI ECL0

IVISCOPE_VAL_ECL1—VXI ECL1

IVISCOPE_VAL_PXI_STAR—PXI Star

IVISCOPE_VAL_RTSL_0—RTSL line 0

IVISCOPE_VAL_RTSL_1—RTSL line 1

IVISCOPE_VAL_RTSL_2—RTSL line 2

IVISCOPE_VAL_RTSL_3—RTSL line 3

IVISCOPE_VAL_RTSL_4—RTSL line 4

IVISCOPE_VAL_RTSL_5—RTSL line 5

IVISCOPE_VAL_RTSL_6—RTSL line 6



Note IVISCOPE_VAL_EXTERNAL—The oscilloscope waits for a trigger on the external trigger input.



tv signal format Pass the type of TV signal on which the oscilloscope triggers. The driver sets the [IviScope TV Trigger Signal Format \[TV\]](#) property to this value.

Defined Values:

IVISCOPE_VAL_NTSCNTSC video signal

IVISCOPE_VAL_PALPAL video signal

IVISCOPE_VAL_SECAMSECAM video signal

Default Value: IVISCOPE_VAL_NTSC

- tv event** Pass the TV event on which you want the oscilloscope to trigger. The driver sets the [IviScope TV Trigger Event \[TV\]](#) property to this value.

Defined Values:

IVISCOPE_VAL_TV_EVENT_FIELD1—Trigger on field 1

IVISCOPE_VAL_TV_EVENT_FIELD2—Trigger on field 2

IVISCOPE_VAL_TV_EVENT_ANY_FIELD—Trigger on any field

IVISCOPE_VAL_TV_EVENT_ANY_LINE—Trigger on any line

IVISCOPE_VAL_TV_EVENT_LINE_NUMBER—Trigger on a line you specify

Default Value: IVISCOPE_VAL_TV_EVENT_FIELD1

-  **Note** If you specify IVISCOPE_VAL_TV_EVENT_LINE_NUMBER, you must set the line number on which the oscilloscope triggers by calling the [IviScope Configure TV Trigger Line Number \[TV\]](#) VI.

- trigger polarity** Pass the polarity of the TV signal. The driver sets the [IviScope TV Trigger Polarity](#) property to this value.

Defined Values:

IVISCOPE_VAL_TV_POSITIVE—Trigger on a positive video sync

IVISCOPE_VAL_TV_NEGATIVE—Trigger on a negative video sync

Default value: IVISCOPE_VAL_TV_POSITIVE

- [error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

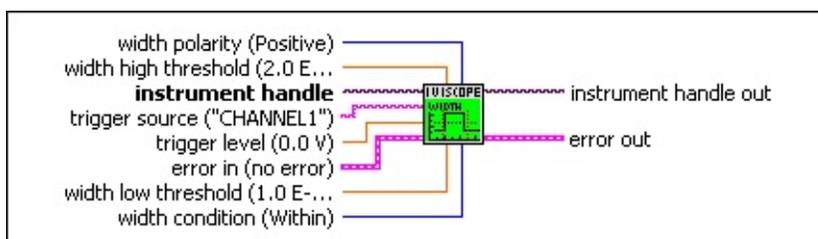
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Configure Width Trigger Source

IviScopeWidthTrigger Capability Group [WT]

This VI configures the width trigger. A width trigger occurs when the oscilloscope detects a positive or negative pulse with a width between, or optionally outside, the width thresholds. You specify the width thresholds with the **Width Low Threshold** and **Width High Threshold** parameters. You specify whether the oscilloscope triggers on pulse widths that are within or outside the width thresholds with the **Width Condition** parameter. You specify the polarity of the pulse with the **Width Polarity** parameter. The trigger does not actually occur until the edge of a pulse that corresponds to the Width Low Threshold, Width High Threshold, Width Condition, and Width Polarity crosses the threshold you specify in the **Trigger Level** parameter.

 **Note** This VI affects instrument behavior only if the trigger type is `IVISCOPE_VAL_WIDTH_TRIGGER`. Call the [IviScope Configure Trigger](#) and [IviScope Configure Trigger Coupling](#) VIs to set the trigger type and trigger coupling before calling this VI.



 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **trigger source** Pass the source you want the oscilloscope to monitor for a trigger. The driver sets the [IviScope Trigger Source](#) property to this value.

After you call one of the [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), [IviScope Read Waveform Measurement \[WM\]](#), or [IviScope Initiate Acquisition](#) VIs, the

oscilloscope waits for the trigger from the source you specify in this parameter.

This control accepts one of the valid channel names or additional trigger sources listed in Additional Trigger Sources.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Additional Trigger Sources

IVISCOPE_VAL_EXTERNAL—External

IVISCOPE_VAL_TTL0—PXI TRIG0 or VXI TTL0

IVISCOPE_VAL_TTL1—PXI TRIG1 or VXI TTL1

IVISCOPE_VAL_TTL2—PXI TRIG2 or VXI TTL2

IVISCOPE_VAL_TTL3—PXI TRIG3 or VXI TTL3

IVISCOPE_VAL_TTL4—PXI TRIG4 or VXI TTL4

IVISCOPE_VAL_TTL5—PXI TRIG5 or VXI TTL5

IVISCOPE_VAL_TTL6—PXI TRIG6 or VXI TTL6

IVISCOPE_VAL_TTL7—PXI TRIG7 or VXI TTL7

IVISCOPE_VAL_ECL0—VXI ECL0

IVISCOPE_VAL_ECL1—VXI ECL1

IVISCOPE_VAL_PXI_STAR—PXI Star

IVISCOPE_VAL_RTSL_0—RTSL line 0

IVISCOPE_VAL_RTSL_1—RTSL line 1

IVISCOPE_VAL_RTSL_2—RTSL line 2

IVISCOPE_VAL_RTSL_3—RTSL line 3

IVISCOPE_VAL_RTISI_4—RTSI line 4

IVISCOPE_VAL_RTISI_5—RTSI line 5

IVISCOPE_VAL_RTISI_6—RTSI line 6



Note IVISCOPE_VAL_EXTERNAL—The oscilloscope waits for a trigger on the external trigger input.



trigger level (volts) Pass the voltage threshold you want the oscilloscope to use for width triggering. The driver sets the [IviScope Trigger Level](#) property to this value.

The oscilloscope triggers when the edge of a pulse that corresponds to the Width Low Threshold, Width High Threshold, Width Condition, and Width Polarity crosses the threshold you specify in this parameter.

Units: volts

Default value: 0.0



Note This parameter only affects instrument behavior when you select a channel or the external trigger input as the Trigger Source. You cannot configure the trigger level that the oscilloscope uses for other trigger sources.



width low threshold (seconds) Pass the low width threshold time. The driver sets the [IviScope Width Low Threshold \[WT\]](#) to this value.

Units: seconds



width high threshold (seconds) Pass the high width threshold time. The driver sets the [IviScope Width High Threshold \[WT\]](#) to this value.

Units: seconds



width polarity Pass the polarity of the pulse that you want to trigger the oscilloscope. The driver sets the [IviScope Width Polarity \[WT\]](#) property to this value.

Defined Values:

IVISCOPE_VAL_WIDTH_POSITIVE—The oscilloscope triggers on a positive pulse.

IVISCOPE_VAL_WIDTH_NEGATIVE—The oscilloscope triggers

on a negative pulse.

Default value: `IVISCOPE_VAL_WIDTH_POSITIVE`

- width condition** Pass whether you want a pulse that is within or outside the Width High Threshold and Width Low Threshold to trigger the oscilloscope.

Defined Values:

`IVISCOPE_VAL_WIDTH_WITHIN`—The oscilloscope triggers on pulses that have a width that is less than the Width High Threshold and greater than the Width Low Threshold.

`IVISCOPE_VAL_WIDTH_OUTSIDE`—The oscilloscope triggers on pulses that have a width that is either greater than the Width High Threshold or less than a Width Low Threshold.

Default Value: `IVISCOPE_VAL_WIDTH_WITHIN`

- [error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

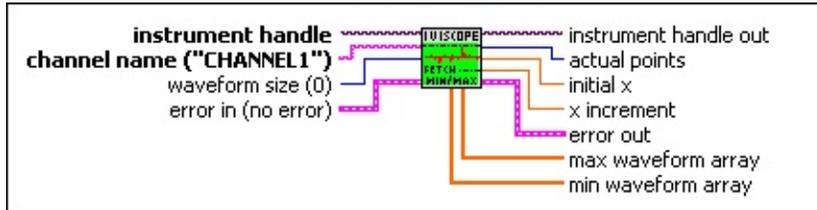
- [error out](#)** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Fetch Min Max Waveform

IviScopeMinMaxWaveform Capability Group [MmW]

This VI returns the minimum and maximum waveforms that the oscilloscope acquires for the channel you specify. If the channel is not enabled for the acquisition, this VI returns the `IVISCOPE_ERROR_CHANNEL_NOT_ENABLED` error. [Details](#)



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- channel name** Pass the virtual channel name that you assign to the instrument in MAX from which you want to fetch the minimum and maximum waveforms.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

- waveform size** Pass the number of elements in the **Min Waveform Array** and **Max Waveform Array** parameters.
- error in (no error)** The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [lviScope Initialize](#) or the [lviScope Initialize With Options](#) VI.



min waveform array Returns the minimum waveform that the oscilloscope acquires.

Units: volts



Note You configure the interpolation method the oscilloscope uses with the [lviScope Configure Interpolation \[I\]](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns IVISCOPE_WARN_INVALID_WFM_ELEMENT. You can test a waveform value for an invalid value condition by calling the [lviScope Is Invalid Waveform Element](#) VI.



max waveform array Returns the maximum waveform that the oscilloscope acquires.

Units: volts



Note You configure the interpolation method the oscilloscope uses with the [lviScope Configure Interpolation \[I\]](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns IVISCOPE_WARN_INVALID_WFM_ELEMENT. You can test a waveform value for an invalid value condition by calling the [lviScope Is Invalid Waveform Element](#) VI.



actual points Indicates the number of points the VI places in the **Min Waveform Array** and **Max Waveform Array** parameters.



initial x Indicates the time of the first point in the Min Waveform

Array and Max Waveform Array. The time is relative to the trigger event. For example, if the oscilloscope acquires the first point in the waveforms 1 second before the trigger, this parameter returns the value -1.0 . If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0 .

Units: seconds

- x increment** Indicates the length of time between points in the Min Waveform Array and Max Waveform Array.

Units: seconds

- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Fetch Min Max Waveform

The waveforms are from an acquisition that you previously initiated. Use this VI to fetch waveforms when you set the acquisition type to `IVISCOPE_VAL_PEAK_DETECT` or `IVISCOPE_VAL_ENVELOPE`. If the acquisition type is not one of the listed types, the VI returns the `IVISCOPE_ERROR_INVALID_ACQ_TYPE` error.

Use the [IviScope Initiate Acquisition](#) VI to start an acquisition on the channels that you enable with the [IviScope Configure Channel](#) VI. The oscilloscope acquires the min/max waveforms for the enabled channels concurrently. You use the [IviScope Acquisition Status](#) VI to determine when the acquisition is complete. You must call this VI separately for each enabled channel to obtain the min/max waveforms.

You can call the [IviScope Read Min Max Waveform \[MmW\]](#) VI instead of the [IviScope Initiate Acquisition](#) VI. The [IviScope Read Min Max Waveform \[MmW\]](#) VI starts an acquisition on all enabled channels, waits for the acquisition to complete, and returns the min/max waveforms for the channel you specify. You call this VI to obtain the min/max waveforms for each of the remaining channels.

Notes

1. After this VI executes, each element in the **Min Waveform Array** and **Max Waveform Array** parameters is either a voltage or a value indicating that the oscilloscope could not sample a voltage.
2. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE defined NaN (Not a Number) value and the VI returns IVISCOPE_WARN_INVALID_WFM_ELEMENT.
3. You can test a waveform value for an invalid value condition by calling the [IviScope Is Invalid Waveform Element](#) VI.
4. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query](#) VI at the conclusion of the sequence.
5. The class driver returns a simulated minimum and maximum waveform when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).

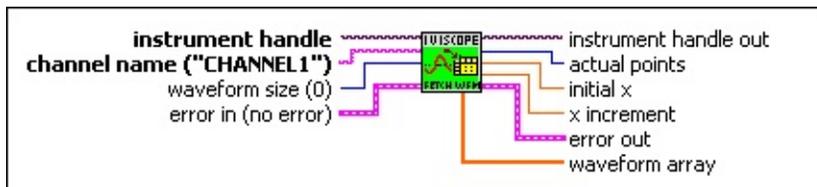
IviScope Fetch Waveform

IviScope Base Capability Group

This VI returns the waveform the oscilloscope acquires for the channel you specify. The waveform is from an acquisition that you initiate prior to calling this VI.

You use the [IviScope Initiate Acquisition](#) VI to start an acquisition on the channels that you enable with the IviScope Configure Channel VI. The oscilloscope acquires waveforms for the enabled channels concurrently. You use the IviScope Acquisition Status VI to determine when the acquisition is complete. You must call this VI separately for each enabled channel to obtain the waveforms.

You can call the [IviScope Read Waveform](#) VI instead of the [IviScope Initiate Acquisition](#) VI. The [IviScope Read Waveform](#) VI starts an acquisition on all enabled channels, waits for the acquisition to complete, and returns the waveform for the channel you specify. You call this VI to obtain the waveform for each of the remaining channels.



Notes

1. After this VI executes, each element in the **Waveform Array** parameter is either a voltage or a value indicating that the oscilloscope could not sample a voltage.
2. You configure the interpolation method the oscilloscope uses with the [IviScope Configure Interpolation \[I\]](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns `IVISCOPE_WARN_INVALID_WFM_ELEMENT`.

3. You can test a waveform value for an invalid value condition by calling the `IviScope Is Invalid Waveform Element VI`.
4. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query VI](#) at the conclusion of the sequence.
5. The class driver returns a simulated waveform when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options VI](#).



channel name Pass the virtual channel name that you assign to the instrument in MAX from which you want to fetch a waveform.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"



waveform size Pass the number of elements in the **Waveform Array** parameter.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



waveform array Returns the waveform that the oscilloscope acquires.

Units: volts



Notes

1. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE defined NaN (Not a Number) value and the VI returns `IVISCOPE_WARN_INVALID_WFM_ELEMENT`.
2. You can test a waveform value for an invalid value condition by calling the [IviScope Is Invalid Waveform Element](#) VI.



actual points Indicates the number of points the VI places in the **Waveform Array** parameter.



initial x Indicates the time of the first point in the Waveform Array. The time is relative to the trigger event. For example, if the oscilloscope acquires the first point in the Waveform Array 1 second before the trigger, this parameter returns the value -1.0 . If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0 .

Units: seconds



x increment Indicates the length of time between points in the Waveform Array.

Units: seconds



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

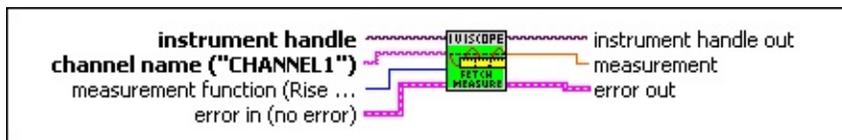
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Fetch Waveform Measurement

IviScopeWaveformMeas Capability Group [WM]

This VI fetches a waveform measurement from the channel you specify. If the channel is not enabled for the acquisition, this VI returns the `IVISCOPE_ERROR_CHANNEL_NOT_ENABLED` error.

The waveform on which the oscilloscope calculates the waveform measurement is from an acquisition that you previously initiated.



Use the [IviScope Initiate Acquisition](#) VI to start an acquisition on the channels that you enable with the IviScope Configure Channel VI. The oscilloscope acquires waveforms for the enabled channels concurrently. You use the IviScope Acquisition Status VI to determine when the acquisition is complete. You call this VI separately for each waveform measurement you want to obtain on a specific channel.

You can call the [IviScope Read Waveform Measurement \[WM\]](#) VI instead of the [IviScope Initiate Acquisition](#) VI. The [IviScope Read Waveform Measurement \[WM\]](#) VI starts an acquisition on all enabled channels. It then waits for the acquisition to complete, obtains a waveform measurement on the channel you specify, and returns the measurement value. You call this VI separately for any other waveform measurement that you want to obtain on a specific channel.



Notes

1. You must configure the appropriate reference levels before you call this VI. You configure the low, mid, and high references either by calling the [IviScope Configure Reference Levels \[WM\]](#) VI or by setting the following properties:

[IviScope Measurement High Reference \[WF\]](#)

[IviScope Measurement Low Reference \[WM\]](#)

[IviScope Measurement Mid Reference \[WM\]](#)

2. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query](#) VI at the conclusion of the sequence.
3. The class driver returns a simulated measurement when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated measurement, refer to [IviScope Simulator Setup Dialog Box](#).



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



channel name Pass the virtual channel name that you assign to the instrument in MAX from which you want to fetch a waveform measurement.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"



measurement function Pass the waveform measurement you want the oscilloscope to perform.

Defined values:

IVISCOPE_VAL_RISE_TIME—Rise Time

The length of time for the first rising edge of the signal to rise from the low reference level to the high reference level.

IVISCOPE_VAL_FALL_TIME—Fall Time

The length of time for the first falling edge of the signal to fall from the high reference level to the low reference level.

IVISCOPE_VAL_FREQUENCY—Frequency

The frequency of one complete cycle in the waveform.

IVISCOPE_VAL_PERIOD—Period

The length of time of one complete cycle in the waveform.

IVISCOPE_VAL_VOLTAGE_RMS—Voltage RMS

The true Root Mean Square voltage of the entire waveform.

IVISCOPE_VAL_VOLTAGE_CYCLE_RMS—Voltage RMS (Cycle)

The true Root Mean Square voltage over the first cycle in the waveform.

IVISCOPE_VAL_VOLTAGE_MAX—Voltage Max

The maximum amplitude found in the entire waveform.

IVISCOPE_VAL_VOLTAGE_MIN—Voltage Min

The minimum amplitude found in the entire waveform.

IVISCOPE_VAL_VOLTAGE_PEAK_TO_PEAK—Voltage Vpp

The absolute difference between the Voltage Max and the Voltage Min.

IVISCOPE_VAL_VOLTAGE_HIGH—Voltage High

The voltage that corresponds to 100% when you use the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the maximum value found. The histogram method uses the most common value found above the middle of the waveform.

IVISCOPE_VAL_VOLTAGE_LOW—Voltage Low

The voltage that corresponds to 0% when you use the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the minimum value found. The histogram method uses the most common value found below the middle of the waveform.

IVISCOPE_VAL_VOLTAGE_AVERAGE—Voltage Avg.

The arithmetic average amplitude of the entire waveform.

IVISCOPE_VAL_VOLTAGE_CYCLE_AVERAGE—Voltage Avg.

(Cycle)

The arithmetic average amplitude of the first cycle in the waveform.

IVISCOPE_VAL_WIDTH_NEG—Width Negative

The length of time between the mid reference level points of the first negative pulse in the waveform.

IVISCOPE_VAL_WIDTH_POS—Width Positive

The length of time between the mid reference level points of the first positive pulse in the waveform.

IVISCOPE_VAL_DUTY_CYCLE_NEG—Duty Cycle Negative

The ratio of the Width Negative to the Period of the first cycle in the waveform expressed as a percentage.

Duty Cycle Negative = $100.0 \times (\text{Width Negative}) / \text{Period}$

IVISCOPE_VAL_DUTY_CYCLE_POS—Duty Cycle Positive

The ratio of the Width Positive to the Period of the first cycle in the waveform expressed as a percentage.

Duty Cycle Positive = $100.0 \times (\text{Width Positive}) / \text{Period}$

IVISCOPE_VAL_AMPLITUDE—Amplitude

The Voltage High less the Voltage Low in Volts over the entire waveform

IVISCOPE_VAL_OVERSHOOT—Overshoot

The relative waveform distortion which follows an edge transition. It is calculated using one of the following formulas:

For the rising edge:

$\text{OVERSHOOT} = ((\text{local maximum} - \text{Voltage High}) / \text{Amplitude}) * 100\%$

where the local maximum is the maximum voltage of the signal in the first half of the time period that commences when the rising edge crosses the high reference level and concludes when the subsequent falling edge crosses the high reference level. For the falling edge:

$\text{OVERSHOOT} = ((\text{Voltage Low} - \text{local minimum}) / \text{Amplitude}) * 100\%$

where the local minimum is the minimum value of the signal measured in the first half of the time period that commences when the falling edge crosses the low reference level and concludes

when the subsequent rising edge crosses the low reference level. The instrument makes the measurement on the edge closest to the beginning of the waveform record.

IVISCOPE_VAL_PRESHOOT—Preshoot

The relative waveform distortion which precedes an edge transition. It is calculated using one of the following formulas:

For the rising edge:

$$\text{PRESHOOT} = ((\text{Voltage Low} - \text{local minimum})/\text{Amplitude}) * 100\%$$

where the local minimum is the minimum value of the signal measured in the second half of the time period that commences when the preceding falling edge crosses the low reference level and concludes when the rising edge crosses the low reference level.

For the falling edge:

$$\text{PRESHOOT} = ((\text{local maximum} - \text{Voltage High})/\text{Amplitude}) * 100\%$$

where the local maximum is the maximum voltage of the signal in the second half of the time period that commences when the preceding rising edge crosses the high reference level and concludes when the falling edge crosses the high reference level. The instrument makes the measurement on the edge closest to the beginning of the waveform record.

Default Value: IVISCOPE_VAL_RISE_TIME



Note You must configure the appropriate reference levels before you call this VI. You configure the low, mid, and high references either by calling the [IviScope Configure Reference Levels \[WM\]](#) VI or by setting the following properties:

[IviScope Measurement High Reference \[WM\]](#)

[IviScope Measurement Low Reference \[WM\]](#)

[IviScope Measurement Mid Reference \[WM\]](#)



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



measurement Returns the measured value.

Units: The units depend on the Measurement Function you select, as shown below.

Rise Time—seconds

Fall Time—seconds

Frequency—hertz

Period—seconds

Voltage RMS—volts

Voltage RMS (Cycle)—volts

Voltage Max—volts

Voltage Min—volts

Voltage Vpp—volts

Voltage High—volts

Voltage Low—volts

Voltage Avg.—volts

Voltage Avg. (Cycle)—volts

Width Negative—seconds

Width Positive—seconds

Duty Cycle Negative—percentage

Duty Cycle Positive—percentage

Amplitude—volts

Overshoot—percentage

Preshoot—percentage



error out The error out cluster passes error or warning information

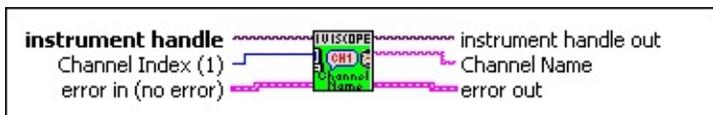
out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScopeGet Channel Name

IviScope Base Capability Group

This VI returns the specific driver defined channel name that corresponds to the one-based index you specify. If the value you pass for the **Index** parameter is less than one or greater than the value of the Channel Count property, the VI returns an empty string in the **Name** parameter and returns an error.



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or [IviScope Initialize With Options](#) VI. The handle identifies a particular instrument session.
- index (1)** A one-based index that defines which name to return. The index must be less than or equal to the [ClassPrefix Channel Count](#) property.
- instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or [IviScope Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

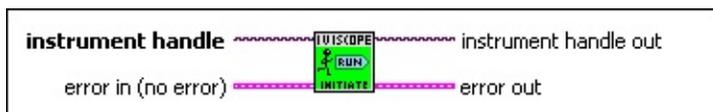
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
- output channel name** Returns the name of the Channel specified by the **index** parameter.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Initiate Acquisition

IviScope Base Capability Group

This VI initiates a waveform acquisition. After you call this VI, the oscilloscope leaves the Idle state and waits for a trigger. The oscilloscope acquires a waveform for each channel you have enabled with the IviScope Configure Channel VI.



Notes

1. This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the [IviScope Error-Query](#) VI at the conclusion of the sequence.
2. This VI performs interchangeability checking when the [IviScope Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviScope Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the help text for the IviScope Interchange Check property.
3. The class driver performs a simulated waveform acquisition when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).

 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if

any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



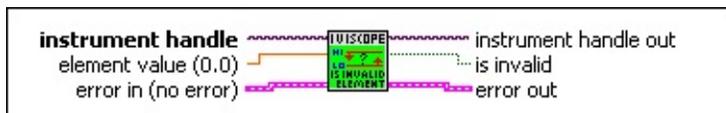
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Is Invalid Waveform Element

IviScope Base Capability Group

After the read and fetch waveform VIs execute, each element in the waveform array contains either a voltage or a value indicating that the oscilloscope could not sample a voltage. The driver uses an IEEE-defined NaN (Not a Number) value to mark as invalid each element in the waveform array for which the oscilloscope could not sample a voltage. This VI determines whether a value you pass from the waveform array is invalid.



- instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- element value** Pass one of the values from the waveform array returned by the read and fetch waveform VIs. The driver uses an IEEE-defined NaN (Not a Number) value to mark as invalid each element in the waveform array for which the oscilloscope could not sample a voltage. The driver determines whether the value you pass is invalid.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.
- is invalid** Returns whether the Element Value is a valid voltage or a value indicating that the oscilloscope could not sample a voltage.

Valid Return Values:

TRUE (1)—The Element Value indicates that the oscilloscope could not sample the voltage.

FALSE (0)—The Element Value is a valid voltage.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

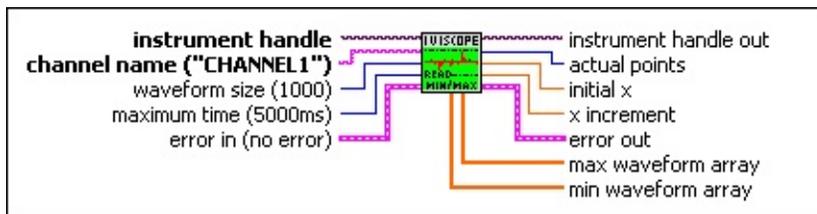
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Read Min Max Waveform

IviScopeMinMaxWaveform Capability Group [MmW]

This VI initiates an acquisition on all channels that you enable with the IviScope Configure Channel VI. If the channel you specify in the **Channel Name parameter** is not enabled for the acquisition, this VI returns the IVISCOPE_ERROR_CHANNEL_NOT_ENABLED error. It then waits for the acquisition to complete and returns the min/max waveforms for the channel you specify. If the oscilloscope did not complete the acquisition within the time period you specify with the **Maximum Time** parameter, the VI returns the IVISCOPE_ERROR_MAX_TIME_EXCEEDED error.

You call the [IviScope Fetch Min Max Waveform \[MmW\]](#) VI to obtain the waveforms for each of the remaining enabled channels without initiating another acquisition.



Notes

1. Use this VI to read waveforms when you set the acquisition type to IVISCOPE_VAL_PEAK_DETECT or IVISCOPE_VAL_ENVELOPE. If the acquisition type is not one of the listed types, the VI returns the IVISCOPE_ERROR_INVALID_ACQ_TYPE error.
2. After this VI executes, each element in the **Min Waveform Array** and **Max Waveform Array** parameters is either a voltage or a value indicating that the oscilloscope could not sample a voltage.
3. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate

points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns IVISCOPE_WARN_INVALID_WFM_ELEMENT.

4. You can test a waveform value for an invalid value condition by calling the IviScope Is Invalid Waveform Element VI.
5. This VI performs interchangeability checking when the [IviScope Interchange Check](#) property is set to TRUE. You use NI Spy to view interchangeability warnings. Alternately, you may use the [IviScope Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviScope Interchange Check](#) property.
6. The class driver returns simulated minimum and maximum waveforms when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).

 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **channel name** Pass the virtual channel name that you assign to the instrument in MAX from which you want to read the minimum and maximum waveforms.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: "CHANNEL1"

waveform size Pass the number of elements in the Min Waveform

Array and Max Waveform Array parameters.

- maximum time (ms)** Pass the maximum length of time in which to allow the read waveform operation to complete.

If the operation does not complete within this time interval, the VI returns the `IVISCOPE_ERROR_MAX_TIME_EXCEEDED` error code. When this occurs, you can call `IviScope Abort` to cancel the read waveform operation and return the oscilloscope to the Idle state.

Units: milliseconds

Defined Values:

`IVISCOPE_VAL_MAX_TIME_INFINITE`

`IVISCOPE_VAL_MAX_TIME_IMMEDIATE`

Default Value: 5000 (ms)

-  **Note** The **Maximum Time** parameter applies only to this VI. It has no effect on other timeout parameters or properties.

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

- min waveform array** Returns the minimum waveform that the oscilloscope acquires.

Units: volts

Notes

1. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the

corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns the `IVISCOPE_WARN_INVALID_WFM_ELEMENT` error.

2. You can test a waveform value for an invalid value condition by calling the IviScope Is Invalid Waveform Element VI.

max waveform array Returns the maximum waveform that the oscilloscope acquires.

Units: volts



Notes

1. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns the `IVISCOPE_WARN_INVALID_WFM_ELEMENT` error.
2. You can test a waveform value for an invalid value condition by calling the IviScope Is Invalid Waveform Element VI.

actual points Indicates the number of points the VI places in the **Min Waveform Array** and **Max Waveform Array** parameters.

initial x Indicates the time of the first point in the Min Waveform Array and Max Waveform Array. The time is relative to the trigger event.

Units: seconds

For example, if the oscilloscope acquires the first point in the waveforms 1 second before the trigger, this parameter returns the value -1.0 . If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0 .

x increment Indicates the length of time between points in the Min and Max Waveform Arrays.

Units: seconds



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

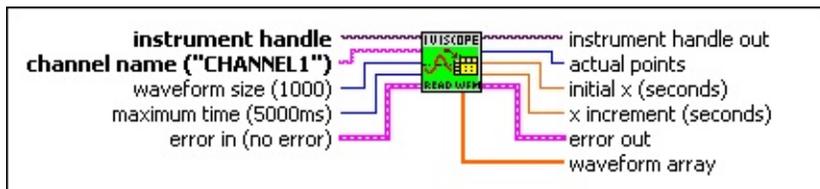
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Read Waveform

IviScope Base Capability Group

This VI initiates an acquisition on all channels that you enable with the [IviScope Configure Channel](#) VI. If the channel you specify in the **Channel Name** parameter is not enabled for the acquisition, this VI returns the `IVISCOPE_ERROR_CHANNEL_NOT_ENABLED` error. It then waits for the acquisition to complete and returns the waveform for the channel you specify. If the oscilloscope did not complete the acquisition within the time period you specify with the **Maximum Time** parameter, the VI returns the `IVISCOPE_ERROR_MAX_TIME_EXCEEDED` error.

You call the [IviScope Fetch Waveform](#) VI to obtain the waveforms for each of the remaining enabled channels without initiating another acquisition.



Notes

1. Use this VI to read waveforms when you set the acquisition mode to `IVISCOPE_VAL_NORMAL`, `IVISCOPE_VAL_HI_RES`, or `IVISCOPE_VAL_AVERAGE`. If the acquisition type is not one of the listed types, the VI returns the `IVISCOPE_ERROR_INVALID_ACQ_TYPE` error.
2. After this VI executes, each element in the **Waveform Array** parameter is either a voltage or a value indicating that the oscilloscope could not sample a voltage.
3. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns the

IVISCOPE_WARN_INVALID_WFM_ELEMENT error.

4. You can test a waveform value for an invalid value condition by calling the [IviScope Is Invalid Waveform Element VI](#).
5. This VI performs interchangeability checking when the [IviScope Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviScope Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviScope Interchange Check](#) property.
6. The class driver returns simulated minimum and maximum waveforms when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).



instrument handle The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



channel name Pass the virtual channel name that you assign to the instrument in MAX from which you want to read a waveform.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""



waveform size Pass the number of elements in the **Waveform Array** parameter.



maximum time (ms) Pass the maximum length of time in which to allow the read waveform operation to complete.

If the operation does not complete within this time interval, the VI returns the IVISCOPE_ERROR_MAX_TIME_EXCEEDED error.

When this occurs, you can call IviScope Abort to cancel the read waveform operation and return the oscilloscope to the Idle state.

Units: milliseconds

Defined Values:

IVISCOPE_VAL_MAX_TIME_INFINITE

IVISCOPE_VAL_MAX_TIME_IMMEDIATE

Default Value: 5000 (ms)



Note The **Maximum Time** parameter applies only to this VI. It has no effect on other timeout parameters or properties.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



waveform array Returns the waveform that the oscilloscope acquires.

Units: volts



Notes

1. You configure the interpolation method the oscilloscope uses with the [IviScope Is Invalid Waveform Element](#) VI. If you disable interpolation, the oscilloscope does not interpolate points in the waveform. If the oscilloscope cannot sample a value for a point in the waveform, the driver sets the corresponding element in the Waveform Array to an IEEE-defined NaN (Not a Number) value and the VI returns the IVISCOPE_WARN_INVALID_WFM_ELEMENT error.
2. You can test a waveform value for an invalid value condition by calling the IviScope Is Invalid Waveform Element VI.



actual points Indicates the number of points the VI places in the

Waveform Array parameter.

- initial x** Indicates the time of the first point in the Waveform Array. The time is relative to the trigger event.

For example, if the oscilloscope acquires the first point in the Waveform Array 1 second before the trigger, this parameter returns the value -1.0 . If the acquisition of the first point occurs at the same time as the trigger, this parameter returns the value 0.0 .

Units: seconds

- x increment** Indicates the length of time between points in the Waveform Array.

Units: seconds

-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

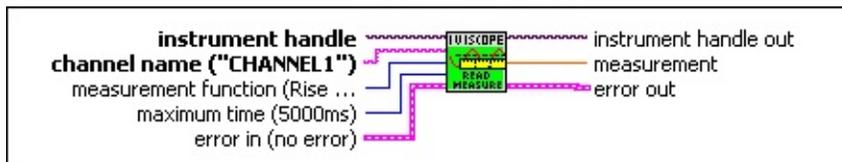
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Read Waveform Measurement

IviScopeWaveformMeas Capability Group [WM]

This VI initiates an acquisition on all channels that you enable with the [IviScope Configure Channel](#) VI. If the channel you specify in the **Channel Name** parameter is not enabled for the acquisition, this VI returns the IVISCOPE_ERROR_CHANNEL_NOT_ENABLED error. It then waits for the acquisition to complete and returns the waveform measurement for the channel you specify. If the oscilloscope did not complete the acquisition within the time period you specify with the **Maximum Time** parameter, the VI returns the IVISCOPE_ERROR_MAX_TIME_EXCEEDED error.

You call the [IviScope Fetch Waveform Measurement \[WM\]](#) VI to obtain any other waveform measurement on a specific channel without initiating another acquisition.



Notes

1. You must configure the appropriate reference levels before you call this VI. You configure the low, mid, and high references either by calling the [IviScope Configure Reference Levels \[WM\]](#) VI or by setting the following properties:
[IviScope Measurement High Reference \[WM\]](#)
[IviScope Measurement Low Reference \[WM\]](#)
[IviScope Measurement Mid Reference \[WM\]](#)
2. This VI performs interchangeability checking when the [IviScope Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviScope Get Next Interchange Warning](#) VI to retrieve

interchangeability warnings. For more information about interchangeability checking, refer to the [IviScope Interchange Check](#) property.

3. The class driver returns simulated minimum and maximum waveforms when this VI is called and the [IviScope Simulate](#) property is set to TRUE and the [IviScope Use Specific Simulation](#) property is set to FALSE. For information on how to configure a simulated waveform, refer to [IviScope Simulator Setup Dialog Box](#).

 **instrument handle** The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.

 **channel name** Pass the virtual channel name that you assign to the instrument in MAX from which you want to read a waveform measurement.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

measurement function Pass the waveform measurement you want the oscilloscope to perform.

Defined values:

IVISCOPE_VAL_RISE_TIME—Rise Time

The length of time for the first rising edge of the signal to rise from the low reference level to the high reference level.

IVISCOPE_VAL_FALL_TIME—Fall Time

The length of time for the first falling edge of the signal to fall from the high reference level to the low reference level.

IVISCOPE_VAL_FREQUENCY—Frequency

The frequency of one complete cycle in the waveform.

IVISCOPE_VAL_PERIOD—Period

The length of time of one complete cycle in the waveform.

IVISCOPE_VAL_VOLTAGE_RMS—Voltage RMS

The true Root Mean Square voltage of the entire waveform.

IVISCOPE_VAL_VOLTAGE_CYCLE_RMS—Voltage RMS (Cycle)

The true Root Mean Square voltage over the first cycle in the waveform.

IVISCOPE_VAL_VOLTAGE_MAX—Voltage Max

The maximum amplitude found in the entire waveform.

IVISCOPE_VAL_VOLTAGE_MIN—Voltage Min

The minimum amplitude found in the entire waveform.

IVISCOPE_VAL_VOLTAGE_PEAK_TO_PEAK—Voltage Vpp

The absolute difference between the Voltage Max and the Voltage Min.

IVISCOPE_VAL_VOLTAGE_HIGH—Voltage High

The voltage that corresponds to 100% when you use the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the maximum value found. The histogram method uses the most common value found above the middle of the waveform.

IVISCOPE_VAL_VOLTAGE_LOW—Voltage Low

The voltage that corresponds to 0% when you use the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the minimum value found. The histogram method uses the most common value found below the middle of the waveform.

IVISCOPE_VAL_VOLTAGE_AVERAGE—Voltage Avg.

The arithmetic average amplitude of the entire waveform.

IVISCOPE_VAL_VOLTAGE_CYCLE_AVERAGE—Voltage Avg. (Cycle)

The arithmetic average amplitude of the first cycle in the waveform.

IVISCOPE_VAL_WIDTH_NEG—Width Negative

The length of time between the mid reference level points of the first negative pulse in the waveform.

IVISCOPE_VAL_WIDTH_POS—Width Positive

The length of time between the mid reference level points of the first positive pulse in the waveform.

IVISCOPE_VAL_DUTY_CYCLE_NEG—Duty Cycle Negative
The ratio of the Width Negative to the Period of the first cycle in the waveform expressed as a percentage.

$$\text{Duty Cycle Negative} = 100.0 \times (\text{Width Negative}) / \text{Period}$$

IVISCOPE_VAL_DUTY_CYCLE_POS—Duty Cycle Positive
The ratio of the Width Positive to the Period of the first cycle in the waveform expressed as a percentage.

$$\text{Duty Cycle Positive} = 100.0 \times (\text{Width Positive}) / \text{Period}$$

IVISCOPE_VAL_AMPLITUDE—Amplitude
The Voltage High less the Voltage Low in Volts over the entire waveform.

IVISCOPE_VAL_OVERSHOOT—Overshoot
The relative waveform distortion which follows an edge transition. It is calculated using one of the following formulas:

For the rising edge:

$$\text{OVERSHOOT} = ((\text{local maximum} - \text{Voltage High}) / \text{Amplitude}) * 100\%$$

where the local maximum is the maximum voltage of the signal in the first half of the time period that commences when the rising edge crosses the high reference level and concludes when the subsequent falling edge crosses the high reference level.

For the falling edge:

$$\text{OVERSHOOT} = ((\text{Voltage Low} - \text{local minimum}) / \text{Amplitude}) * 100\%$$

where the local minimum is the minimum value of the signal measured in the first half of the time period that commences when the falling edge crosses the low reference level and concludes when the subsequent rising edge crosses the low reference level. The instrument makes the measurement on the edge closest to the beginning of the waveform record.

IVISCOPE_VAL_PRESHOOT—Preshoot
The relative waveform distortion which precedes an edge

transition. It is calculated using one of the following formulas:

For the rising edge:

$$\text{PRESHOOT} = ((\text{Voltage Low} - \text{local minimum})/\text{Amplitude}) * 100\%$$

where the local minimum is the minimum value of the signal measured in the second half of the time period that commences when the preceding falling edge crosses the low reference level and concludes when the rising edge crosses the low reference level.

For the falling edge:

$$\text{PRESHOOT} = ((\text{local maximum} - \text{Voltage High})/\text{Amplitude}) * 100\%$$

where the local maximum is the maximum voltage of the signal in the second half of the time period that commences when the preceding rising edge crosses the high reference level and concludes when the falling edge crosses the high reference level. The instrument makes the measurement on the edge closest to the beginning of the waveform record.

Default Value: IVISCOPE_VAL_RISE_TIME



Note You must configure the appropriate reference levels before you call this VI. You configure the low, mid, and high references either by calling the [IviScope Configure Reference Levels \[WM\]](#) VI or by setting the following properties:

[IviScope Measurement High Reference \[WM\]](#)

[IviScope Measurement Low Reference \[WM\]](#)

[IviScope Measurement Mid Reference \[WM\]](#)



maximum time (ms) Pass the maximum length of time in which to allow the read waveform measurement operation to complete.

If the operation does not complete within this time interval, the VI returns the IVISCOPE_ERROR_MAX_TIME_EXCEEDED error code. When this occurs, you can call IviScope Abort to cancel the read waveform operation and return the oscilloscope to the Idle state.

Units: milliseconds.

Defined Values:

IVISCOPE_VAL_MAX_TIME_INFINITE
IVISCOPE_VAL_MAX_TIME_IMMEDIATE

Default Value: 5000 (ms)



Note The **Maximum Time** parameter applies only to this VI. It has no effect on other timeout parameters or properties.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviScope Initialize](#) or the [IviScope Initialize With Options](#) VI.



measurement Returns the measured value.

Units The units depend on the Measurement Function you select, as shown below:

Rise Time—seconds

Fall Time—seconds

Frequency—hertz

Period—seconds

Voltage RMS—volts

Voltage RMS (Cycle)—volts

Voltage Max—volts

Voltage Min—volts

Voltage Vpp—volts

Voltage High—volts

Voltage Low—volts

Voltage Avg.—volts

Voltage Avg. (Cycle)—volts

Width Negative—seconds

Width Positive—seconds

Duty Cycle Negative—percentage

Duty Cycle Positive—percentage

Amplitude—volts

Overshoot—percentage

Preshoot—percentage



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviScope Properties

Expand this book to view an alphabetized list of IviScope properties.

IviScope AC Line Trigger Slope [AT]

IviScopeAcLineTrigger [AT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure AC Line Trigger Slope

Description

Specifies the slope of the zero crossing upon which the scope triggers. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_AC_LINE_TRIGGER.

Property Node Path

IviScope»Trigger»AC Line Triggering [AT]»AC Line Trigger Slope [AT]

Defined Values

[IVISCOPE_VAL_AC_LINE_EITHER](#)
[IVISCOPE_VAL_AC_LINE_POSITIVE](#)
[IVISCOPE_VAL_AC_LINE_NEGATIVE](#)

IviScope Acquisition Start Time

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Acquisition Record

Description

This properties specifies the length of time from the trigger event to the first point in the waveform record. The units are seconds. If the value is positive, the first point in the waveform record occurs after the trigger event. If the value is negative, the first point in the waveform record occurs before the trigger event.

Property Node Path

IviScope»Acquisition»Acquisition Start Time

IviScope Acquisition Type

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Acquisition Type

Description

Specifies how the oscilloscope acquires data and fills the waveform record. When you set this property to `IVISCOPE_VAL_ENVELOPE` or `IVISCOPE_VAL_PEAK_DETECT`, the oscilloscope acquires minimum and maximum waveforms. To retrieve the minimum and maximum waveforms, use the [IviScope Read Min Max Waveform \[MmW\]](#) and [IviScope Fetch Min Max Waveform \[MmW\]](#) VIs.

Property Node Path

IviScope»Acquisition»Acquisition Type

Defined Values

[IVISCOPE_VAL_NORMAL](#)

[IVISCOPE_VAL_PEAK_DETECT](#)

[IVISCOPE_VAL_HI_RES](#)

[IVISCOPE_VAL_ENVELOPE](#)

[IVISCOPE_VAL_AVERAGE](#)

IviScope Channel Count

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of channels available on the instrument.

Property Node Path

IviScope»Inherent IVI Settings»Specific Driver Capabilities»Channel Count

IviScope Channel Enabled

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	IviScope Configure Channel

Description

This channel-based property specifies whether the oscilloscope acquires a waveform for a channel when you call the [IviScope Initiate Acquisition](#), [IviScope Read Waveform](#), [IviScope Read Min Max Waveform \[MmW\]](#), or [IviScope Read Waveform Measurement](#) VI.

Property Node Path

IviScope»Channel»Channel Enabled

IviScope Glitch Condition [GT]

IviScopeGlitchTrigger [GT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Glitch Trigger Source

Description

Specifies the glitch condition that triggers the oscilloscope. The glitch trigger occurs when the oscilloscope detects a pulse with a width less than or greater than the width value you specify with the [IviScope Glitch Width \[GT\]](#) property. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_GLITCH_TRIGGER.

Property Node Path

IviScope»Trigger»Glitch Triggering [GT]»Glitch Condition [GT]

Defined Values

[IVISCOPE_VAL_GLITCH_LESS_THAN](#)

[IVISCOPE_VAL_GLITCH_GREATER_THAN](#)

IviScope Glitch Polarity [GT]

IviScopeGlitchTrigger Property [GT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Glitch Trigger Source

Description

Specifies the polarity of the glitch that triggers the oscilloscope. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_GLITCH_TRIGGER.

Property Node Path

IviScope»Trigger»Glitch Triggering [GT]»Glitch Polarity [GT]

Defined Values

[IVISCOPE_VAL_GLITCH_POSITIVE](#)

[IVISCOPE_VAL_GLITCH_NEGATIVE](#)

[IVISCOPE_VAL_GLITCH_EITHER](#)

IviScope Glitch Width [GT]

IviScopeGlitchTrigger Property [GT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Glitch Trigger Source

Description

Specifies the glitch width. The units are seconds. The oscilloscope triggers when it detects a pulse with a width that is less than the value you specify. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_GLITCH_TRIGGER.

Property Node Path

IviScope»Trigger»Glitch Triggering [GT]»Glitch Width [GT]

IviScope Horizontal Minimum Number of Points

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Acquisition Record

Description

Specifies the minimum number of points you require in the waveform record for each channel. The instrument driver uses the value you specify to configure the record length that the oscilloscope uses for waveform acquisition. The [IviScope Horizontal Record Length](#) property returns the actual record length.

Property Node Path

IviScope»Acquisition»Horizontal Minimum Number of Points

IviScope Horizontal Record Length

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviScope Actual Record Length

Description

Returns the actual number of points the oscilloscope acquires for each channel. The value is equal to or greater than the minimum number of points you specify [IviScope Horizontal Minimum Number of Points](#) property.

Allocate a array of this size or greater to pass as the **Waveform Array** parameter of the [IviScope Read Waveform](#) and [IviScope Fetch Waveform](#) VIs.



Note Oscilloscopes may use different size records depending on the value specified for the [IviScope Acquisition Type](#) property.

Property Node Path

IviScope»Acquisition»Horizontal Record Length

IviScope Horizontal Sample Rate

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviScope Actual Sample Rate

Description

Returns the effective digitizing rate using the current configuration. The units are samples per second.

Property Node Path

IviScope»Acquisition»Horizontal Sample Rate

IviScope Horizontal Time Per Record

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	Up	IviScope Configure Acquisition Record

Description

Specifies the length of time in seconds that corresponds to the record length.

Property Node Path

IviScope»Acquisition»Horizontal Time Per Record

IviScope Initiate Continuous [CA]

IviScopeContinuousAcquisition Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Initiate Continuous

Description

Specifies whether the oscilloscope continuously initiates waveform acquisition. If you set this property to TRUE, the oscilloscope immediately waits for another trigger after the previous waveform acquisition is complete. Setting this property to TRUE is useful when you require continuous updates of the oscilloscope display.

Property Node Path

IviScope»Acquisition»Initiate Continuous [CA]

IviScope Input Impedance

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	IviScope Configure Channel Characteristics

Description

This channel-based property specifies the input impedance for the channel. The units are ohms.

Property Node Path

IviScope»Channel»Input Impedance

IviScope Interpolation [I]

IviScopeInterpolation Property [I]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Interpolation

Description

Specifies the interpolation method the oscilloscope uses when it cannot sample a voltage for every point in the waveform record.

Property Node Path

IviScope»Acquisition»Interpolation [I]

Defined Values

[IVISCOPE_VAL_NO_INTERPOLATION](#)

[IVISCOPE_VAL_SINE_X](#)

[IVISCOPE_VAL_LINEAR](#)

IviScope Maximum Input Frequency

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	Up	IviScope Configure Channel Characteristics

Description

This channel-based property specifies the maximum input frequency 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 (Hz).

Property Node Path

IviScope»Channel»Maximum Input Frequency

IviScope Measurement High Reference [WM]

IviScopeMeasWaveform Property [WM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Reference Levels

Description

Specifies the high reference the oscilloscope uses for waveform measurements. The value is a percentage of the difference between the Voltage High and Voltage Low. The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.

Property Node Path

IviScope»Waveform Measurement [WM]»Measurement High Reference [WM]

IviScope Measurement Low Reference

IviScopeMeasWaveform Property [WM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Reference Levels

Description

Specifies the low reference the oscilloscope uses for waveform measurements. The value is a percentage of the difference between the Voltage High and Voltage Low. The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.

Property Node Path

IviScope»Waveform Measurement [WM]»Measurement Low Reference [WM]

IviScope Measurement Mid Reference [WM]

IviScopeMeasWaveform Property [WM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Reference Levels

Description

Specifies the mid reference the oscilloscope uses for waveform measurements. The value is a percentage of the difference between the Voltage High and Voltage Low. The oscilloscope calculates the Voltage High and the Voltage Low using either the min/max or histogram methods. The min/max method uses the maximum and minimum values found. The histogram method uses the most common values found above and below the middle of the waveform.

Property Node Path

IviScope»Waveform Measurement [WM]»Measurement Mid Reference [WM]

IviScope Number of Averages [AA]

IviScopeAverageAcquisition Property [AA]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Number of Averages

Description

Specifies the number of waveform the oscilloscope acquires and averages. After the oscilloscope acquires as many waveforms as this property specifies, it returns to the Idle state. This property affects instrument behavior only when the [IviScope Acquisition Type](#) property is set to [IVISCOPE_VAL_AVERAGE](#).

Property Node Path

IviScope»Acquisition»Number of Averages [AA]

IviScope Number of Envelopes [MmW]

IviScopeMinMaxWaveform Property [MmW]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Number of Envelopes

Description

When you set the [IviScope Acquisition Type](#) property to [IVISCOPE_VAL_ENVELOPE](#), the oscilloscope acquires multiple waveforms. After each waveform acquisition, the oscilloscope keeps the minimum and maximum values it finds for each point in the waveform record. This property specifies the number of waveforms the oscilloscope acquires and analyzes to create the minimum and maximum waveforms. After the oscilloscope acquires as many waveforms as this property specifies, it returns to the Idle state. This property affects instrument operation only when the [IviScope Acquisition Type](#) property is set to [IVISCOPE_VAL_ENVELOPE](#).

Property Node Path

IviScope»Acquisition»Number of Envelopes [MmW]

IviScope Probe Attenuation

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	IviScope Configure Channel

Description

This channel-based property specifies the scaling factor by which the probe you attach to the channel attenuates the input. For example, when you use a 10:1 probe, set this property to 10.0.

This driver reserves negative values to control the oscilloscope's automatic probe sense capability. Setting this property to `IVISCOPE_VAL_PROBE_SENSE_ON` configures the oscilloscope to sense the attenuation of the probe automatically. After you enable the automatic probe sense, subsequent queries of this property return the value `IVISCOPE_VAL_PROBE_SENSE_ON`. Use the [IviScope Probe Sense Value \[PAS\]](#) property to obtain the actual probe attenuation.

If you set the oscilloscope to sense the probe attenuation automatically, the probe attenuation value can change at any time. When the oscilloscope detects a new probe attenuation value, other settings in the oscilloscope might also change. The driver has no way of knowing when these changes occur. Therefore, when you enable the automatic probe sense capability, this driver disables caching for properties that depend on the probe attenuation. These properties include [IviScope Vertical Range](#), [IviScope Vertical Offset](#), and all the properties that configure trigger levels. To maximize performance, set this property to a manual probe attenuation setting.

If the oscilloscope is set to sense the probe attenuation automatically, setting this property to a positive value disables the automatic probe sense and configures the oscilloscope to use the manual probe attenuation you specify.

If you use a manual probe attenuation, you must set this property to reflect the new probe attenuation each time you attach a different probe.

Property Node Path

IviScope»Channel»Attenuation

Defined Values

IVISCOPE_VAL_PROBE_SENSE_ON

IviScope Probe Sense Value

IviScopeProbeAutoSense Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	IviScope Auto Probe Sense Value

Description

This channel-based property returns the probe attenuation value the oscilloscope automatically senses. If you disable the automatic probe sense capability, this property returns the manual probe attenuation setting.

Property Node Path

IviScope»Channel»Probe Sense Value

IviScope Runt High Threshold [RT]

IviScopeRuntTrigger Property [RT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Runt Trigger Source

Description

Specifies the high threshold the oscilloscope uses for runt triggering. The units are volts. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_RUNT_TRIGGER.

Property Node Path

IviScope»Trigger»Runt Triggering [RT]»Runt High Threshold [RT]

IviScope Runt Low Threshold [RT]

IviScopeRuntTrigger Property [RT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Runt Trigger Source

Description

Specifies the low threshold the oscilloscope uses for runt triggering. The units are volts. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_RUNT_TRIGGER.

Property Node Path

IviScope»Trigger»Runt Triggering [RT]»Runt Low Threshold [RT]

IviScope Runt Polarity [RT]

IviScopeRuntTrigger Property [RT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Runt Trigger Source

Description

Specifies the polarity of the runt that triggers the oscilloscope. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_RUNT_TRIGGER.

Property Node Path

IviScope»Trigger»Runt Triggering [RT]»Runt Polarity [RT]

Defined Values

[IVISCOPE_VAL_RUNT_POSITIVE](#)

[IVISCOPE_VAL_RUNT_NEGATIVE](#)

[IVISCOPE_VAL_RUNT_EITHER](#)

IviScope Sample Mode [SM]

IviScopeSampleMode Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviScope Actual Sample Mode

Description

Returns the sample mode the oscilloscope is currently using.

Property Node Path

IviScope»Acquisition»Sample Mode [SM]

Defined Values

[IVISCOPE_VAL_REAL_TIME](#)

[IVISCOPE_VAL_EQUIVALENT_TIME](#)

IviScope Trigger Coupling

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Trigger Coupling

Description

Specifies how the oscilloscope couples the trigger source.

Property Node Path

IviScope»Trigger»Trigger Coupling

Defined Values

[IVISCOPE_VAL_AC](#)

[IVISCOPE_VAL_DC](#)

[IVISCOPE_VAL_HF_REJECT](#)

[IVISCOPE_VAL_LF_REJECT](#)

[IVISCOPE_VAL_NOISE_REJECT](#)

IviScope Trigger Holdoff

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	See Note Below	IviScope Configure Trigger



Note Many scopes have a small, non-zero value as the minimum value for this property. To configure the instrument to use the shortest trigger holdoff, you can specify a value of zero for this property.

Description

Specifies the length of time the oscilloscope waits after it detects a trigger until the oscilloscope enables the trigger subsystem to detect another trigger. The units are seconds.

The [IviScope Trigger Holdoff](#) property affects instrument operation only when the oscilloscope requires multiple acquisitions to build a complete waveform. The oscilloscope requires multiple waveform acquisitions when it uses equivalent-time sampling or when you set the [IviScope Acquisition Type](#) property to IVISCOPE_VAL_ENVELOPE or IVISCOPE_VAL_AVERAGE.

Property Node Path

IviScope»Trigger»Trigger Holdoff

IviScope Trigger Level

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Edge Trigger Source IviScope Configure Glitch Trigger Source IviScope Configure Width Trigger Source

Description

Specifies the voltage threshold for the trigger subsystem. The units are volts. This property affects instrument behavior only when you set the [IviScope Trigger Type](#) to IVISCOPE_VAL_EDGE_TRIGGER, IVISCOPE_VAL_GLITCH_TRIGGER, or IVISCOPE_VAL_WIDTH_TRIGGER.

Property Node Path

IviScope»Trigger»Trigger Level

IviScope Trigger Modifier [TM]

IviScopeTriggerModifier Property [TM]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Trigger Modifier

Description

Specifies the trigger modifier. The trigger modifier determines the oscilloscope's behavior in the absence of the trigger you configure.

Property Node Path

IviScope»Trigger»Trigger Modifier [TM]

Defined Values

[IVISCOPE_VAL_NO_TRIGGER_MOD](#)

[IVISCOPE_VAL_AUTO](#)

[IVISCOPE_VAL_AUTO_LEVEL](#)

IviScope Trigger Slope

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Edge Trigger Source

Description

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

This property effects instrument operation only when the [IviScope Trigger Type](#) property is set to [IVISCOPE_VAL_EDGE_TRIGGER](#).

Property Node Path

IviScope»Trigger»Trigger Slope

Defined Values

[IVISCOPE_VAL_POSITIVE](#)

[IVISCOPE_VAL_NEGATIVE](#)

IviScope Trigger Source

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Edge Trigger Source IviScope Configure TV Trigger Source IviScope Configure Runt Trigger Source IviScope Configure Glitch Trigger Source IviScope Configure Width Trigger Source

Description

Specifies the source the oscilloscope monitors for a trigger. Set this property to a channel name or to one of the defined values.

Property Node Path

IviScope»Trigger»Trigger Source

Defined Values

[IVISCOPE_VAL_EXTERNAL](#)

[IVISCOPE_VAL_TTL0](#)

[IVISCOPE_VAL_TTL1](#)

[IVISCOPE_VAL_TTL2](#)

[IVISCOPE_VAL_TTL3](#)

[IVISCOPE_VAL_TTL4](#)

[IVISCOPE_VAL_TTL5](#)

[IVISCOPE_VAL_TTL6](#)

[IVISCOPE_VAL_TTL7](#)

[IVISCOPE_VAL_ECL0](#)

IVISCOPE_VAL_ECL1

IVISCOPE_VAL_PXI_STAR

IVISCOPE_VAL_RTSI_0

IVISCOPE_VAL_RTSI_1

IVISCOPE_VAL_RTSI_2

IVISCOPE_VAL_RTSI_3

IVISCOPE_VAL_RTSI_4

IVISCOPE_VAL_RTSI_5

IVISCOPE_VAL_RTSI_6

IviScope Trigger Type

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Trigger

Description

This property specifies the trigger type.

Property Node Path

IviScope»Trigger»Trigger Type

Defined Values

[IVISCOPE_VAL_EDGE_TRIGGER](#)

[IVISCOPE_VAL_WIDTH_TRIGGER](#)

[IVISCOPE_VAL_RUNT_TRIGGER](#)

[IVISCOPE_VAL_GLITCH_TRIGGER](#)

[IVISCOPE_VAL_TV_TRIGGER](#)

[IVISCOPE_VAL_IMMEDIATE_TRIGGER](#)

[IVISCOPE_VAL_AC_LINE_TRIGGER](#)

IviScope TV Trigger Event [TV]

IviScopeTVTrigger Property [TV]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure TV Trigger Source

Description

Specifies the event on which the oscilloscope triggers. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_TV_TRIGGER.

Property Node Path

IviScope»Trigger»TV Triggering [TV]»TV Trigger Event [TV]

Defined Values

[IVISCOPE_VAL_TV_EVENT_FIELD1](#)

[IVISCOPE_VAL_TV_EVENT_FIELD2](#)

[IVISCOPE_VAL_TV_EVENT_ANY_FIELD](#)

[IVISCOPE_VAL_TV_EVENT_ANY_LINE](#)

[IVISCOPE_VAL_TV_EVENT_LINE_NUMBER](#)

IviScope TV Trigger Line Number [TV]

IviScopeTVTrigger Property [TV]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure TV Trigger Source

Description

Specifies the line on which the oscilloscope triggers. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_TV_TRIGGER and when the [IviScope TV Trigger Event \[TV\]](#) property is set to IVISCOPE_VAL_LINE_NUMBER. The line number setting is independent of the field. This means that to trigger on the first line of the second field, you must set this property to the value of 263 (assuming that field one had 262 lines).

Property Node Path

IviScope»Trigger»TV Triggering [TV]»TV Trigger Line Number [TV]

IviScope TV Trigger Polarity [TV]

IviScopeTVTrigger Property [TV]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure TV Trigger Source

Description

Specifies the polarity of the TV signal. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_TV_TRIGGER.

Property Node Path

IviScope»Trigger»TV Triggering [TV]»TV Trigger Polarity [TV]

Defined Values

[IVISCOPE_VAL_TV_POSITIVE](#)

[IVISCOPE_VAL_TV_NEGATIVE](#)

IviScope TV Trigger Signal Format [TV]

IviScopeTVTrigger Property [TV]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure TV Trigger Source

Description

Specifies the format of the TV signal on which the oscilloscope triggers. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_TV_TRIGGER.

Property Node Path

IviScope»Trigger»TV Triggering [TV]»TV Trigger Signal Format [TV]

Defined Values

[IVISCOPE_VAL_NTSC](#)

[IVISCOPE_VAL_PAL](#)

[IVISCOPE_VAL_SECAM](#)

IviScope Vertical Coupling

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	IviScope Configure Channel

Description

This channel-based property specifies how the oscilloscope couples the input signal for the channel.

Property Node Path

IviScope»Channel»Vertical Coupling

Defined Values

[IVISCOPE_VAL_AC](#)

[IVISCOPE_VAL_DC](#)

[IVISCOPE_VAL_GND](#)

IviScope Vertical Offset

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	IviScope Configure Channel

Description

This channel-based property specifies the location of the center of the range that you specify with the [IviScope Vertical Range](#) property. Express the value in volts and 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.

Property Node Path

IviScope»Channel»Vertical Offset

IviScope Vertical Range

IviScope Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	Up	IviScope Configure Channel

Description

This channel-based property specifies the absolute value of the input range the oscilloscope can acquire for the channel. The units are volts. For example, to acquire a sine wave which spans between -5.0 and 5.0 V, you set this property to 10.0 V.

Property Node Path

IviScope»Channel»Vertical Range

IviScope Width Condition [WT]

IviScopeWidthTrigger Property [WT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Width Trigger Source

Description

Specifies whether a pulse that is within or outside the high and low thresholds triggers the oscilloscope. You specify the high and low thresholds with the [IviScope Width High Threshold \[WT\]](#) and [IviScope Width Low Threshold \[WT\]](#) properties. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_WIDTH_TRIGGER.

Property Node Path

IviScope»Trigger Subsystem»Width Triggering [WT]»Width Condition [WT]

Defined Values

[IVISCOPE_VAL_WIDTH_WITHIN](#)

[IVISCOPE_VAL_WIDTH_OUTSIDE](#)

IviScope Width High Threshold [WT]

IviScopeWidthTrigger Property [WT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Width Trigger Source

Description

Specifies the high width threshold time. The units are seconds. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_WIDTH_TRIGGER.

Property Node Path

IviScope»Trigger»Width Triggering [WT]»Width High Threshold [WT]

IviScope Width Low Threshold [WT]

IviScopeWidthTrigger Property [WT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Width Trigger Source

Description

Specifies the low width threshold time. The units are seconds. This property affects instrument operation only when the [IviScope Trigger Type](#) property is set to IVISCOPE_VAL_WIDTH_TRIGGER.

Property Node Path

IviScope»Trigger»Width Triggering [WT]»Width Low Threshold [WT]

IviScope Width Polarity [WT]

IviScopeWidthTrigger Property [WT]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviScope Configure Width Trigger Source

Description

Specifies the polarity of the pulse that triggers the oscilloscope. This property affects instrument operation only when the [viScope Trigger Type](#) property is set to IVISCOPE_VAL_WIDTH_TRIGGER.

Property Node Path

IviScope»Trigger»Width Triggering [WT]»Width Polarity [WT]

Defined Values

[IVISCOPE_VAL_WIDTH_POSITIVE](#)

[IVISCOPE_VAL_WIDTH_NEGATIVE](#)

IviScope Values

Expand this book to view an alphabetized list of IviScope values.

IVISCOPE_VAL_AC_LINE_EITHER

Configures the oscilloscope to trigger on either positive or negative slope zero crossings of the network supply voltage.

Defined Value: 3

IVISCOPE_VAL_AC_LINE_NEGATIVE

Configures the oscilloscope to trigger on negative slope zero crossings of the network supply voltage.

Defined Value: 2

IVISCOPE_VAL_AC_LINE_POSITIVE

Configures the oscilloscope to trigger on positive slope zero crossings of the network supply voltage.

Defined Value: 1

IVISCOPE_VAL_AC_LINE_TRIGGER

Configures the oscilloscope for AC line triggering. An AC Line trigger occurs when the oscilloscope detects a positive zero crossing, negative zero crossing, or optionally either positive or negative zero crossing of the network supply voltage.

Defined Value: 7

IVISCOPE_VAL_AC_TRIGGER

The oscilloscope AC couples the trigger signal.

Defined Value: 0

IVISCOPE_VAL_AC_Trigger_Coupling

The oscilloscope AC couples the trigger signal.

Defined Value: 0

IVISCOPE_VAL_AC_VERTICAL_COUPLING

The oscilloscope AC couples the input signal.

Defined Value: 0

IVISCOPE_VAL_ACQ_COMPLETE

Defined Value: 1

IVISCOPE_VAL_ACQ_IN_PROGRESS

Defined Value: 0

IVISCOPE_VAL_ACQ_STATUS_UNKNOWN

Defined Value: -1

IVISCOPE_VAL_AUTO

The oscilloscope automatically triggers if the configured trigger does not occur within the oscilloscope's timeout period.

Defined Value: 2

IVISCOPE_VAL_AUTO_LEVEL

The oscilloscope adjusts the trigger level if the trigger you specify does not occur within the oscilloscope's timeout period.

Defined Value: 3

IVISCOPE_VAL_AVERAGE

Configures the oscilloscope to acquire multiple waveforms and calculate the average value for each point in the waveform record. You specify the number of waveforms to acquire with the [IVISCOPE_ATTR_NUM_AVERAGES](#) property. The oscilloscope uses real-time or equivalent time sampling.

Defined Value: 4

IVISCOPE_VAL_DC_TRIGGER

The oscilloscope DC couples the trigger signal.

Defined Value: 1

IVISCOPE_VAL_DC_Trigger_Coupling

The oscilloscope DC couples the trigger signal.

Defined Value: 1

IVISCOPE_VAL_DC_VERTICAL_COUPLING

The oscilloscope DC couples the input signal.

Defined Value: 1

IVISCOPE_VAL_ECL0

The oscilloscope waits until it receives a trigger on the ECL0 line.

Defined Value: "VAL_ECL0"

IVISCOPE_VAL_ECL1

The oscilloscope waits until it receives a trigger on the ECL1 line.

Defined Value: "VAL_ECL1"

IVISCOPE_VAL_EDGE_TRIGGER

Configures the oscilloscope for edge triggering. An edge trigger occurs when the trigger signal passes through the voltage threshold specified by the [IVISCOPE_ATTR_TRIGGER_LEVEL](#) property and has the slope specified by the [IVISCOPE_ATTR_TRIGGER_SLOPE](#) property.

Defined Value: 1

IVISCOPE_VAL_ENVELOPE

Sets the oscilloscope to the envelope acquisition mode. The oscilloscope acquires multiple waveforms and keeps the minimum and maximum voltages it acquires for each point in the waveform record. You specify the number of waveforms the oscilloscope acquires with the [IVISCOPE_ATTR_NUM_ENVELOPES](#) property. The oscilloscope can use real-time or equivalent-time sampling.

Defined Value: 3

IVISCOPE_VAL_EQUIVALENT_TIME

Indicates that the oscilloscope is using equivalent time sampling.

Defined Value: 1

IVISCOPE_VAL_EXTERNAL

The oscilloscope waits for a trigger on the external trigger input.

Defined Value: "VAL_EXTERNAL"

IVISCOPE_VAL_GLITCH_EITHER

The oscilloscope triggers on either a positive or negative glitch.

Defined Value: 3

IVISCOPE_VAL_GLITCH_GREATER_THAN

The oscilloscope triggers when the pulse width is greater than the value you specify with the IVISCOPE_ATTR_GLITCH_WIDTH property.

Defined Value: 2

IVISCOPE_VAL_GLITCH_LESS_THAN

The oscilloscope triggers when the pulse width is less than the value you specify with the IVISCOPE_ATTR_GLITCH_WIDTH attribute.

Defined Value: 1

IVISCOPE_VAL_GLITCH_NEGATIVE

The oscilloscope triggers on a negative glitch.

Defined Value: 2

IVISCOPE_VAL_GLITCH_POSITIVE

The oscilloscope triggers on a positive glitch.

Defined Value: 1

IVISCOPE_VAL_GLITCH_TRIGGER

Configures the oscilloscope for glitch triggering. A glitch trigger occurs when the trigger signal has a pulse with a width that is less than the glitch width. You specify the glitch width with the

IVISCOPE_ATTR_GLITCH_WIDTH property. You specify the polarity of the pulse with the IVISCOPE_ATTR_GLITCH_POLARITY property. The trigger does not actually occur until the edge of a pulse that corresponds to the glitch width and polarity you specify crosses the trigger level that you specify with the IVISCOPE_ATTR_TRIGGER_LEVEL property.

Defined Value: 4

IVISCOPE_VAL_GND

The oscilloscope ground couples the input signal.

Defined Value: 2

IVISCOPE_VAL_HF_REJECT

The oscilloscope filters out the low frequencies from the trigger signal.

Defined Value: 3

IVISCOPE_VAL_HI_RES

Configures the oscilloscope to oversample the input signal. The oscilloscope calculates the average value that corresponds to each position in the waveform record. The oscilloscope uses only real-time sampling.

Defined Value: 2

IVISCOPE_VAL_IMMEDIATE_TRIGGER

The oscilloscope does not wait for a trigger of any kind.

Defined Value: 6

IVISCOPE_VAL_LF_REJECT

The oscilloscope filters out the high frequencies from the trigger signal.

Defined Value: 4

IVISCOPE_VAL_LINEAR

The oscilloscope uses a linear approximation to interpolate a value when it can not resolve a voltage in the waveform record.

Defined Value: 3

IVISCOPE_VAL_MAX_TIME_IMMEDIATE

The function returns immediately. If no valid measurement value exists, the function returns an error.

Defined Value: 0x0

IVISCOPE_VAL_MAX_TIME_INFINITE

The function waits indefinitely for the measurement to complete.

Defined Value: 0xFFFFFFFFUL

IVISCOPE_VAL_NEGATIVE

A negative (falling) edge passing through the trigger level triggers the oscilloscope.

Defined Value: 0

IVISCOPE_VAL_NO_INTERPOLATION

The oscilloscope does not interpolate points in the waveform. Instead, the driver sets every element in the waveform array for which the oscilloscope cannot resolve a value to an IEEE-defined NaN (Not a Number) value.

Defined Value: 1

IVISCOPE_VAL_NO_TRIGGER_MOD

The oscilloscope waits until the trigger you specify occurs.

Defined Value: 1

IVISCOPE_VAL_NOISE_REJECT

The oscilloscope filters out the noise from the trigger signal.

Defined Value: 5

IVISCOPE_VAL_NORMAL

Configures the oscilloscope to acquire one sample for each point in the waveform record. The oscilloscope uses real-time or equivalent time sampling.

Defined Value: 0

IVISCOPE_VAL_NTSC

Sets the TV trigger signal format to NTSC.

Defined Value: 1

IVISCOPE_VAL_PAL

Sets the TV trigger signal format to PAL.

Defined Value: 2

IVISCOPE_VAL_PEAK_DETECT

Sets the oscilloscope to the peak-detect acquisition mode. The oscilloscope oversamples the input signal and keeps the minimum and maximum values that correspond to each position in the waveform record. The oscilloscope uses only real-time sampling.

Defined Value: 1

IVISCOPE_VAL_POSITIVE

A positive (rising) edge passing through the trigger level triggers the oscilloscope.

Defined Value: 1

IVISCOPE_VAL_PROBE_SENSE_ON

Setting this property to [IVISCOPE_VAL_PROBE_SENSE_ON](#) configures the oscilloscope to sense the attenuation of the probe automatically. After you enable the automatic probe sense, subsequent queries of this property return the value [IVISCOPE_VAL_PROBE_SENSE_ON](#). Use the [IVISCOPE_ATTR_PROBE_SENSE_VALUE](#) property to obtain the actual probe attenuation.

If you set the oscilloscope to sense the probe attenuation automatically, the probe attenuation value can change at any time. When the oscilloscope detects a new probe attenuation value, other settings in the oscilloscope might also change. The driver has no way of knowing when these changes occur. Therefore, when you enable the automatic probe sense capability, the driver disables caching for properties that depend on the probe attenuation. These properties include [IVISCOPE_ATTR_VERTICAL_RANGE](#), [IVISCOPE_ATTR_VERTICAL_OFFSET](#), and all the properties that configure trigger levels. To maximize performance, you should set this property to a manual probe attenuation setting.

If the oscilloscope is set to sense the probe attenuation automatically, setting this property to a positive value disables the automatic probe sense and configures the oscilloscope to use the manual probe attenuation you specify.

Defined Value: -1

IVISCOPE_VAL_PXI_STAR

The oscilloscope waits until it receives a trigger on the PXI Star bus.

Defined Value: "VAL_PXI_STAR"

IVISCOPE_VAL_REAL_TIME

Indicates that the oscilloscope is using real-time sampling.

Defined Value: 0

IVISCOPE_VAL_RTISI_0

The oscilloscope waits until it receives a trigger on the RTSI 0 line.

Defined Value: "VAL_RTISI_0"

IVISCOPE_VAL_RTSI_1

The oscilloscope waits until it receives a trigger on the RTSI 1 line.

Defined Value: "VAL_RTSI_1"

IVISCOPE_VAL_RTISI_2

The oscilloscope waits until it receives a trigger on the RTSI 2 line.

Defined Value: "VAL_RTISI_2"

IVISCOPE_VAL_RTISI_3

The oscilloscope waits until it receives a trigger on the RTSI 3 line.

Defined Value: "VAL_RTISI_3"

IVISCOPE_VAL_RTISI_4

The oscilloscope waits until it receives a trigger on the RTSI 4 line.

Defined Value: "VAL_RTISI_4"

IVISCOPE_VAL_RTISI_5

The oscilloscope waits until it receives a trigger on the RTSI 5 line.

Defined Value: "VAL_RTISI_5"

IVISCOPE_VAL_RTISI_6

The oscilloscope waits until it receives a trigger on the RTSI 6 line.

Defined Value: "VAL_RTISI_6"

IVISCOPE_VAL_RUNT_EITHER

The oscilloscope triggers on either a positive or negative runt.

Defined Value: 3

IVISCOPE_VAL_RUNT_NEGATIVE

The oscilloscope triggers on a negative runt. A negative runt occurs when a falling edge crosses the high runt threshold and does not cross the low runt threshold before re-crossing the high runt threshold.

Defined Value: 2

IVISCOPE_VAL_RUNT_POSITIVE

The oscilloscope triggers on a positive runt. A positive runt occurs when a rising edge crosses the low runt threshold and does not cross the high runt threshold before re-crossing the low runt threshold.

Defined Value: 1

IVISCOPE_VAL_RUNT_TRIGGER

Configures the oscilloscope for runt triggering. A runt trigger occurs when the trigger signal crosses one of the runt thresholds twice without crossing the other runt threshold. You specify the runt thresholds with the IVISCOPE_ATTR_RUNT_HIGH_THRESHOLD and IVISCOPE_ATTR_RUNT_LOW_THRESHOLD properties. You specify the polarity of the runt with the IVISCOPE_ATTR_RUNT_POLARITY properties.

Defined Value: 3

IVISCOPE_VAL_SECAM

Sets the TV trigger signal format to SECAM.

Defined Value: 3

IVISCOPE_VAL_SINE_X

The oscilloscope uses a $\text{sine}(x)/x$ calculation to interpolate a value when it can not resolve a voltage in the waveform record.

Defined Value: 2

IVISCOPE_VAL_TTL0

The oscilloscope waits until it receives a trigger on the TTL0 line.

Defined Value: "VAL_TTL0"

IVISCOPE_VAL_TTL1

The oscilloscope waits until it receives a trigger on the TTL1 line.

Defined Value: "VAL_TTL1"

IVISCOPE_VAL_TTL2

The oscilloscope waits until it receives a trigger on the TTL2 line.

Defined Value: "VAL_TTL2"

IVISCOPE_VAL_TTL3

The oscilloscope waits until it receives a trigger on the TTL3 line.

Defined Value: "VAL_TTL3"

IVISCOPE_VAL_TTL4

The oscilloscope waits until it receives a trigger on the TTL4 line.

Defined Value: "VAL_TTL4"

IVISCOPE_VAL_TTL5

The oscilloscope waits until it receives a trigger on the TTL5 line.

Defined Value: "VAL_TTL5"

IVISCOPE_VAL_TTL6

The oscilloscope waits until it receives a trigger on the TTL6 line.

Defined Value: "VAL_TTL6"

IVISCOPE_VAL_TTL7

The oscilloscope waits until it receives a trigger on the TTL7 line.

Defined Value: "VAL_TTL7"

IVISCOPE_VAL_TV_EVENT_ANY_FIELD

Sets the oscilloscope to trigger on any field.

Defined Value: 3

IVISCOPE_VAL_TV_EVENT_ANY_LINE

Sets the oscilloscope to trigger on any line.

Defined Value: 4

IVISCOPE_VAL_TV_EVENT_FIELD1

Sets the oscilloscope to trigger on field 1 of the video signal.

Defined Value: 1

IVISCOPE_VAL_TV_EVENT_FIELD2

Sets the oscilloscope to trigger on field 2 of the video signal.

Defined Value: 2

IVISCOPE_VAL_TV_EVENT_LINE_NUMBER

Sets the oscilloscope to trigger on a specific line number you specify with the IVISCOPE_ATTR_TV_TRIGGER_LINE_NUMBER property.

Defined Value: 5

IVISCOPE_VAL_TV_FIELD1

Field 1 of the video signal.

Defined Value: 1

IVISCOPE_VAL_TV_FIELD2

Field 2 of the video signal.

Defined Value: 2

IVISCOPE_VAL_TV_NEGATIVE

Negative video sync pulse.

Defined Value: 2

IVISCOPE_VAL_TV_POSITIVE

Positive video sync pulse.

Defined Value: 1

IVISCOPE_VAL_TV_TRIGGER

Configures the oscilloscope to trigger on TV signals. You configure the trigger with the [IVISCOPE_TV_TRIGGER_SIGNAL_FORMAT](#), [IVISCOPE_TV_TRIGGER_LINE_NUMBER](#), and [IVISCOPE_ATTR_TV_TRIGGER_POLARITY](#) properties.

Defined Value: 5

IVISCOPE_VAL_WIDTH_NEGATIVE

Configures the oscilloscope to trigger on negative pulses that have a width that meets the condition you specify with the [IVISCOPE_ATTR_WIDTH_CONDITION](#) property.

Defined Value: 2

IVISCOPE_VAL_WIDTH_OUTSIDE

Configures the oscilloscope to trigger on pulses that have a width that is either greater than the high threshold or less than a low threshold. You specify the high and low thresholds with the

IVISCOPE_ATTR_WIDTH_HIGH_THRESHOLD and IVISCOPE_ATTR_WIDTH_LOW_THRESHOLD properties.

Defined Value: 2

IVISCOPE_VAL_WIDTH_POSITIVE

Configures the oscilloscope to trigger on positive pulses that have a width that meets the condition you specify with the [IVISCOPE_ATTR_WIDTH_CONDITION](#) property.

Defined Value: 1

IVISCOPE_VAL_WIDTH_TRIGGER

Configures the oscilloscope for width triggering. A width trigger occurs when the oscilloscope detects a positive or negative pulse with a width between, or optionally outside, the width thresholds. You specify the width thresholds with the [IVISCOPE_ATTR_WIDTH_HIGH_THRESHOLD](#) and [IVISCOPE_ATTR_WIDTH_LOW_THRESHOLD](#) properties. You specify whether the oscilloscope triggers on pulse widths that are within or outside the width thresholds with the [IVISCOPE_ATTR_WIDTH_CONDITION](#) property. You specify the polarity of the pulse with the [IVISCOPE_ATTR_WIDTH_POLARITY](#) property. The trigger does not actually occur until the edge of a pulse that corresponds to the width thresholds, width condition, and polarity you specify crosses the trigger level that you specify with the [IVISCOPE_ATTR_TRIGGER_LEVEL](#) property.

Defined Value: 2

IVISCOPE_VAL_WIDTH_WITHIN

Configures the oscilloscope to trigger on pulses that have a width that is less than the high threshold and greater than the low threshold. You specify the high and low thresholds with the [IVISCOPE_ATTR_WIDTH_HIGH_THRESHOLD](#) and [IVISCOPE_ATTR_WIDTH_LOW_THRESHOLD](#) properties.

Defined Value: 1

Properties for Controlling IviScope Simulation



Note You can set Simulation Attributes in MAX on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** or in the simulation interactive panels. Refer to **National Instruments IVI Driver Help»Configuring Your System** for more information.

Two sets of properties exist for use with the IviScope Simulation Driver. The following table describes properties that control behavior of the driver. Another table lists [properties that simulate the status](#) of specific driver functions.

An [IviScope Simulator Setup Dialog Box](#) exists to help you configure the measurement simulation for the IviScope Simulator driver.

Properties for Controlling IviScope Simulation

Name	Data Type	Access	Applies to	Description
INTERACTIVE_SIMULATION	Boolean	W	No	Specifies whether driver to interact in interactive mode simulation driver user interface panel getting information. Non-interactive when you do not use interactive panel in your test program.
SELF_TEST_CODE	Integer	W	No	Specifies the self-test code. When the IviScope Self-Test VI is called, the self test code parameter returns this value.
SELF_TEST_MSG	String	W	No	Specifies the self-test message. When Self-Test VI is called, the test message parameter returns this string.
ERROR_QUERY_CODE	Integer	W	No	Specifies the error query code. When the IviScope Query VI is called, the query code parameter returns this value.
ERROR_QUERY_MSG	String	W	No	Specifies the error query message. When Error-Query VI is called, the error message parameter returns this string.
DRIVER_REV_QUERY	String	W	No	Specifies the driver revision message. When IviScope Revision VI is called, the instance parameter returns this string.

				revision param this string.
INSTR_REV_QUERY	String	W	No	Specifies the fir message. When Revision Query the firmware re parameter retur
WAVEFORM	Int32	W	Yes	Specifies the w simulate at the channel. Defined Values: 1—Sine 2—Square 3—Triangle 4—Ramp-up 5—Ramp-down 6—DC
FREQUENCY	Real64	W	Yes	Specifies the fre waveform to sin selected chann property does n simulation beha NISSCOPE_ATT property is set t Range: 100.0 to
AMPLITUDE	Real64	W	Yes	Specifies the pe amplitude of the simulate at the channel. This p not affect simul when the NISSCOPE_ATT property is set t Valid Range: 50.0E-3 to 10.0

DC_OFFSET	Real64	W	Yes	Specifies the DC offset of the waveform to simulate for the selected channel. Valid Range: -5.0 to 5.0 V
DUTY_CYCLE	Real64	W	Yes	Specifies the duty cycle of the waveform to simulate for a selected channel. This property does not affect the simulation behavior of the channel. This property is set to 50.0 percent. Valid Range: 20.0 to 80.0 percent
PHASE_SHIFT	Real64	W	Yes	Specifies the phase shift of the waveform to simulate for a selected channel. This property does not affect the simulation behavior of the channel. This property is set to 0.0 degrees. Valid Range: 0.0 to 360.0 degrees
SNR	Real64	W	Yes	Specifies the signal-to-noise ratio to simulate for a selected channel. The signal-to-noise ratio determines the amount of noise to simulate in the waveform acquisition. Valid Range: 0.001 to 100.0
SIMULATE_STATUS_CODE	Boolean	W	No	Specifies whether to return error codes from driver functions. For more information, see the IviScope Properties Simulation table.

Related Topics

[Setting Up Simulated Errors](#)

Setting Up Simulation for VXi*plug&play* VIs

IviScope Simulator Setup Dialog Box

[Two sets of properties](#) exist for use with the IviScope simulation driver. You use the dialog box shown below to configure the measurement simulation for the IviScope simulation driver. Scroll down to see a description of each control.



Always prompt for output data simulation

Leaving this control selected causes the panel to appear at every signal acquisition event in your program. All simulated acquisitions in the program use the same waveform configuration information. Unselecting this control causes the waveform simulation panel to never appear again during the course of your program.

Channel

This control specifies the channel for which you want to configure a simulation waveform.

Waveform

This control selects the waveform to use for the simulated signal.

Frequency

This control selects the frequency of the waveform to simulate at the selected channel. This value does not affect simulation behavior when the waveform is set to DC. Valid Range: 100.0 to 15.0E6 Hz

Amplitude

This control selects the peak-to-peak amplitude of the waveform to simulate at the selected channel. This value does not affect simulation behavior when the waveform is set to DC. Valid Range: 50.0E-3 to 10.0 Vpp

DC Offset

This control selects the DC offset of the waveform to simulate at the selected channel. Valid Range: -5.0 to 5.0 V

Phase Shift

This control selects the starting phase of the waveform to simulate at the selected channel. This value does not affect simulation behavior when the waveform is set to DC. Valid Range: 0.0 to 360.0 degrees

SNR

This control selects the signal-to-noise ratio of the simulated signal. The SNR determines the amount of noise to introduce into the waveform generation. This option allows you to simulate less than perfect waveforms. Valid Range: 0.001 to +inf

Properties for Status Simulation in IviScope

When an IviScope driver function is called, the driver queries whether `NISSCOPE_ATTR_SIMULATE_STATUS_CODE` is enabled. If enabled, the driver gets the appropriate function's simulation status code. If the value of the simulation status code represents a warning (has a positive value), then it is return as the function's return status only if no other error or warning occurred before the function ends. If the value of the simulation status code represents an error (has a negative value), then it is returned as the function's return status only if no other error occurred before the function ends.

The following IviScope VIs do not support status code simulation:

- IviScope Get Error
- IviScope Clear Error
- IviScope Get Next Interchange Warning
- IviScope Reset Next Interchange Warning
- IviScope Clear Next Interchange Warning
- IviScope Get Next Coercion Record

Each VI supported by the IviScope class driver has a corresponding property that determines the status code to return when status code simulation is enabled. The following list indicates the status code properties and the VI for which they return a value. These properties are all of type `ViInt32`, non-readable, and non-channel based:



Note In the following table, the literal string `NISSCOPE_ATTR_` precedes all property names.

IviScope Properties for Status Simulation

Property	VI
INIT_STATUS	IviScope Initialize , IviScope Initialize With Options
CLOSE_STATUS	IviScope Close
RESET_STATUS	IviScope Reset
SELF_TEST_STATUS	IviScope Self-Test
ERROR_QUERY_STATUS	IviScope Error-Query
ERROR_MESSAGE_STATUS	IviScope Error Message
REVISION_QUERY_STATUS	IviScope Revision Query
RESET_DEFAULT_STATUS	IviScope Reset With Defaults
DISABLE_STATUS	IviScope Disable
INVALIDATE_STATUS	IviScope Invalidate All Attributes
GET_CH_NAME_STATUS	IviScope Get Channel Name
CONFIGURE_CHANNEL_STATUS	IviScope Configure Channel
CONFIGURE_CHAN_CHARACTERISTICS_STATUS	IviScope Configure Channel Characteristics
CONFIGURE_ACQUISITION_TYPE_STATUS	IviScope Configure Acquisition Type

CONFIGURE_ACQUISITION_RECORD__STATUS	IviScope Configure Acquisition Record
ACTUAL_RECORD_LENGTH_STATUS	IviScope Actual Record Length
CONFIGURE_TRIGGER_STATUS	IviScope Configure Trigger
CONFIGURE_TRIGGER_COUPLING_STATUS	IviScope Configure Trigger Coupling
CONFIGURE_EDGE_TRIGGER_SOURCE_STATUS	IviScope Configure Edge Trigger Source
READ_WAVEFORM_STATUS	IviScope Read Waveform
ABORT_STATUS	IviScope Abort
INITIATE_ACQUISITION_STATUS	IviScope Initiate Acquisition
ACQUISITION_STATUS_STATUS	IviScope Acquisition Status
FETCH_WAVEFORM_STATUS	IviScope Fetch Waveform
IS_INVALID_WFM_ELEMENT_STATUS	IviScope Is Invalid Waveform Element
CONFIGURE_TV_TRIGGER_SOURCE_STATUS	IviScope Configure TV Trigger Source
CONFIGURE_TV_TRIGGER_LINE_NUMBER_STATUS	IviScope Configure TV Trigger Line Number
CONFIGURE_RUNT_TRIGGER_SOURCE_STATUS	IviScope Configure Runt

	Trigger Source
CONFIGURE_GLITCH_TRIGGER_SOURCE_STATUS	IviScope Configure Glitch Trigger Source
CONFIGURE_WIDTH_TRIGGER_SOURCE_STATUS	IviScope Configure Width Trigger Source
CONFIGURE_AC_LINE_TRIGGER_SLOPE_STATUS	IviScope Configure AC Line Trigger Slope
CONFIGURE_TRIGGER_MODIFIER_STATUS	IviScope Configure Trigger Modifier
CONFIGURE_NUM_ENVELOPES_STATUS	IviScope Configure Number of Envelopes
READ_MIN_MAX_WAVEFORM_STATUS	IviScope Read Min Max Waveform
FETCH_MIN_MAX_WAVEFORM_STATUS	IviScope Fetch Min Max Waveform
CONFIGURE_REF_LEVELS_STATUS	IviScope Configure Reference Levels
READ_WAVEFORM_MEASUREMENT_STATUS	IviScope Read Waveform Measurement
FETCH_WAVEFORM_MEASUREMENT_STATUS	IviScope Fetch Waveform Measurement
CONFIGURE_NUM_AVERAGES_STATUS	IviScope Configure

	<u>Number of Averages</u>
CONFIGURE_INITIATE_CONTINUOUS_STATUS	<u>IviScope Configure Initiate Continuous</u>
CONFIGURE_INTERPOLATION_STATUS	<u>IviScope Configure Interpolation</u>
SAMPLE_MODE_STATUS	<u>IviScope Actual Sample Mode</u>
SAMPLE_RATE_STATUS	<u>IviScope Actual Sample Rate</u>
AUTO_PROBE_SENSE_VALUE_STATUS	<u>IviScope Auto Probe Sense Value</u>
AUTO_SETUP_STATUS	<u>IviScope Auto Setup</u>



IviSpecAn Class Driver Reference for LabVIEW

IviSpecAn VI and Property Reference

[VI Tree](#)

[Properties by Group](#)

[Error and Completion Codes](#)

IviSpecAn Class Driver Overview

This instrument driver provides programming support for the IviSpecAn Class. The IviSpecAn class is designed to support the typical swept analyzer as well as common extended functionality found in instruments that are more complex. The IviSpecAn class conceptualizes a swept analyzer as an instrument that can measure the amplitude at discrete frequencies while sweeping a receiver through a range of frequencies. Typically, the measured quantity is a voltage or power level. The driver contains all the VIs required in the IviSpecAn specification defined by the IVI Foundation. This driver requires NI-VISA and the IVI Compliance Package to be installed.

There is a second type of analyzer that is often referred to as a dynamic signal analyzer, an audio analyzer, or an FFT analyzer. While many of the end results are the same, the process of taking the measurements is quite different. Since they are so different, they are not covered in this help topic.

The IviSpecAn class is divided into the base capability group and several extension groups. The base capability group is used to configure an analyzer for a typical sweep, initiate that sweep, and return an array of measured values.

In addition to the base capabilities, the IviSpecAn class defines extended capabilities for spectrum analyzers that have:

- Simple mathematical operations on multiple traces
- Marker and delta-marker functionality
- External and Video trigger sources
- Basic display function of specifying units per division
- External mixer capability to expand the frequency range
- Peak preselector to reduce noise and increase dynamic range

The IviSpecAn extended capabilities are arranged into a set of extension capability groups.

IviSpecAn Extension Capability Groups

The IviSpecAn class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviSpecAnMultitrace</u>	Defines extensions for analyzers capable of performing simple mathematical VIs on one or more traces.
<u>IviSpecAnMarker</u>	Supports spectrum analyzers that have markers. Markers are applied to traces and used for a wide range of operations. Some operations are simple, such as reading an amplitude value at an X-axis position, while others operations are complex, such as signal tracking
<u>IviSpecAnTrigger</u>	Specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state.
<u>IviSpecAnExternalTrigger</u>	Specifies the external trigger level and external trigger slope when the Trigger Source property is set to external, which causes the analyzer to leave the Wait-For-Trigger state.
<u>IviSpecAnSoftwareTrigger</u>	Supports spectrum analyzers that can acquire traces based on a software trigger signal. You can send a software trigger to cause signal output to occur. Affects instrument behavior when the Trigger Source property is set to Software.
<u>IviSpecAnVideoTrigger</u>	Specifies the video trigger level and video trigger slope when the Trigger Source

	property is set to Video, which causes the analyzer to leave the Wait-For-Trigger state.
<u>IviSpecAnDisplay</u>	Controls the display related properties.
<u>IviSpecAnMarkerType</u>	Supports analyzers that have multiple marker types.
<u>IviSpecAnDeltaMarker</u>	Supports analyzers that have delta-marker capabilities. A delta marker has the same properties as a normal marker except that its position and amplitude are relative to a fixed reference point. This reference point is defined when the marker is converted from a normal marker to a delta marker.
<u>IviSpecAnExternalMixer</u>	Allows the use of an analyzer to measure values that are outside of the normal frequency range of the equipment. When using an external mixer, many of the settings of the analyzer have to be carefully converted to allow you to know what is meant by the values read.
<u>IviSpecAnPreselector</u>	Controls preselectors. Preselectors are a network of filters and preamplifiers that are built into one unit for reducing noise and increasing dynamic range of an analyzer. Preselectors are often separate instruments, but they are instruments that only work with spectrum analyzers. Some analyzers have internal preselectors.

Related Topic

[IVI Inherent Class Capabilities](#)

IviSpecAn Extension Capability Groups

The IviSpecAn class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
<u>IviSpecAnMultitrace</u>	Defines extensions for analyzers capable of performing simple mathematical VIs on one or more traces.
<u>IviSpecAnMarker</u>	Supports spectrum analyzers that have markers. Markers are applied to traces and used for a wide range of operations. Some operations are simple, such as reading an amplitude value at an X-axis position, while others operations are complex, such as signal tracking
<u>IviSpecAnTrigger</u>	Specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state.
<u>IviSpecAnExternalTrigger</u>	Specifies the external trigger level and external trigger slope when the Trigger Source property is set to external, which causes the analyzer to leave the Wait-For-Trigger state.
<u>IviSpecAnSoftwareTrigger</u>	Supports spectrum analyzers that can acquire traces based on a software trigger signal. You can send a software trigger to cause signal output to occur. Affects instrument behavior when the Trigger Source property is set to Software.
<u>IviSpecAnVideoTrigger</u>	Specifies the video trigger level and video trigger slope when the Trigger Source

IviSpecAnDisplay

property is set to Video, which causes the analyzer to leave the Wait-For-Trigger state.

Controls the display related properties.

IviSpecAnMarkerType

Supports analyzers that have multiple marker types.

IviSpecAnDeltaMarker

Supports analyzers that have delta-marker capabilities. A delta marker has the same properties as a normal marker except that its position and amplitude are relative to a fixed reference point. This fixed reference point is defined when the marker is converted from a normal marker to a delta marker.

IviSpecAnExternalMixer

Allows the use of an analyzer to measure values that are outside of the normal frequency range of the equipment. When using an external mixer, many of the settings of the analyzer have to be carefully converted to allow the user to know what is meant by the values read.

IviSpecAnPreselector

Controls preselectors. Preselectors are a network of filters and preamplifiers that are built into one unit for reducing noise and increasing dynamic range of an analyzer. Preselectors are often separate instruments, but they are instruments that only work with spectrum analyzers. Some analyzers have internal preselectors.

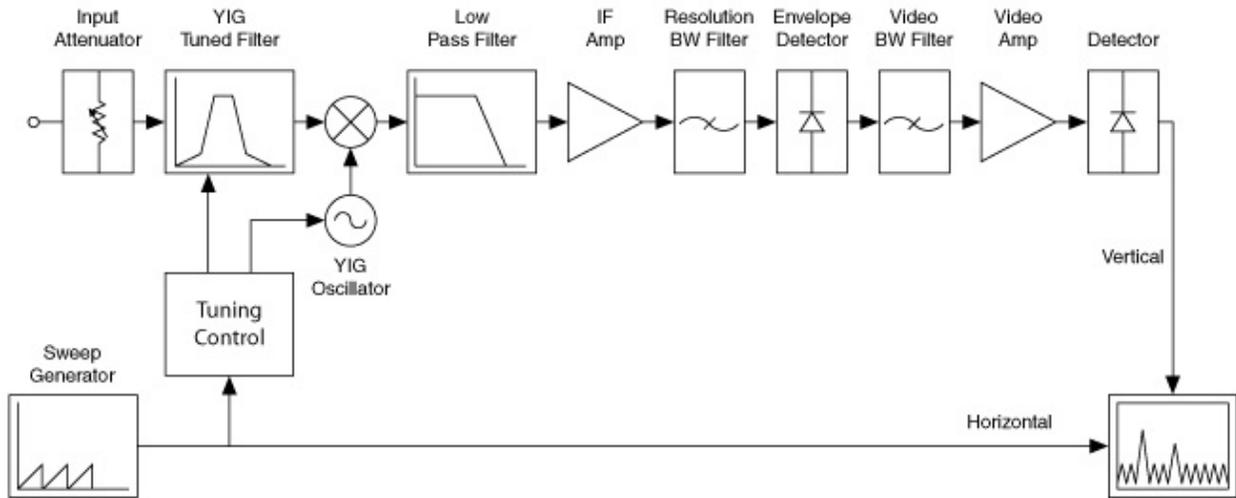
IviSpecAnBase Capability Group Overview

The IviSpecAnBase capability group supports spectrum analyzers that configure and take a frequency sweep. A frequency sweep is thought of as adjusting the frequency of a tuner from the start frequency to the stop frequency in a defined amount of time. For more information about time domain measurements, refer to [Time Domain Measurements](#). While the tuner is being adjusted, power levels (or voltage levels) are being measured for the frequencies. The result is an array of amplitude versus frequency data. In addition to configuring the start and stop frequency, a user can also set a frequency offset. This affects the setting of the spectrum analyzer's absolute frequencies, such as start, stop, center, and marker. It does not affect values, which are the difference of frequencies, such as span and delta marker.

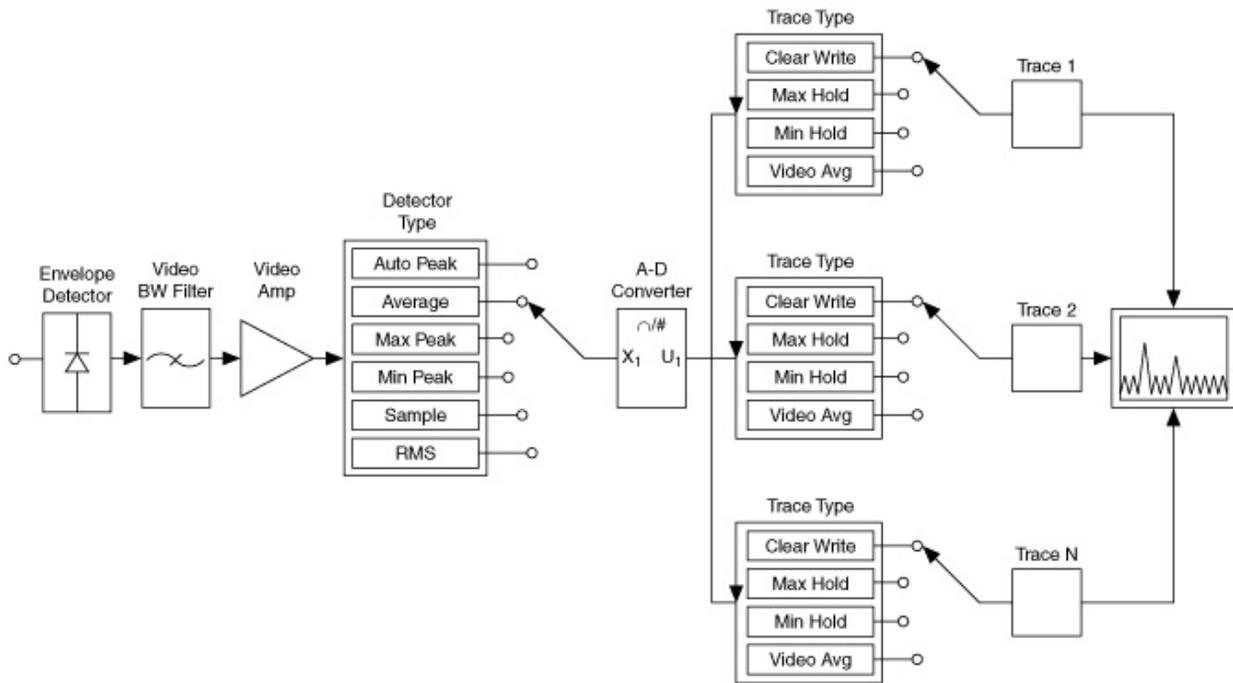
The IviSpecAnBase capability group also includes VIs for configuring the analyzer as well as initiating and retrieving measurements. This includes configuring the sweep, range, and resolution. A typical spectrum analyzer does not have several input channels. It can, however, have several traces. Multiple traces are covered in the IviSpecAnMultitrace extension group. The trigger VIs are covered in Trigger, External Trigger, and Video Trigger extension groups.

In addition, most spectrum analyzers have a coupled function mode, which links the resolution bandwidth (RBW), video bandwidth (VBW), and sweep time together. For more information about this behavior, refer to [Sweep Coupling Overview](#).

The following figure shows the tuning, filtering, and detection process in a typical spectrum analyzer.



The following figure shows how the detector detects the signal and outputs the signal on multiple traces.



Sweep Coupling Overview

Many spectrum analyzers are capable of coupling the resolution bandwidth, video bandwidth, and sweep time together. The instrument makes decisions based on the various settings to ensure the accuracy of the measurement. If you elect to choose the settings, it is possible to place constraints on the system that make accurate measurements impossible. Most analyzers respond to this condition with a message indicating that the measurement is uncalibrated.

When all the settings are coupled, the instrument makes decisions along the following hierarchy. The RBW and VBW are typically locked together and set for 1% of the span. For example, if the span were set for 100 MHz, the instrument chooses 1 MHz for the RBW and VBW. To measure very low signal levels, the RBW may need to be more narrowed, which slows the sweep speed. Lastly, the sweep time is dependent on the RBW and the VBW. As a rule of thumb:

$$\text{Sweep Time} * \text{Span} / \text{RBW}^2$$

As can be seen from the above equation, the RBW has a drastic effect on sweep speed. These are issues to keep in mind when developing automated tests.

When considering interchangeability and measurement accuracy, it is important to consider the way different instruments couple settings together. The valid combination of RBW, VBW, and sweep time on one spectrum analyzer may not be a valid combination on another spectrum analyzer. Awareness of these differences ensures that the results obtained in one system correlate with the results from another system.

IviSpecAnBase Functions

[IviSpecAn_Abort](#)

[IviSpecAn_AcquisitionStatus](#)

[IviSpecAn_ConfigureAcquisition](#)

[IviSpecAn_ConfigureFrequencyCenterSpan](#)

[IviSpecAn_ConfigureFrequencyOffset](#)

[IviSpecAn_ConfigureFrequencyStartStop](#)

[IviSpecAn_ConfigureLevel](#)

[IviSpecAn_ConfigureSweepCoupling](#)

[IviSpecAn_ConfigureTraceType](#)

[IviSpecAn_FetchYTrace](#)

[IviSpecAn_GetTraceName](#)

[IviSpecAn_Initiate](#)

[IviSpecAn_QueryTraceSize](#)

[IviSpecAn_ReadYTrace](#)

IviSpecAnBase Attributes

[IVISPECAN_ATTR_AMPLITUDE_UNITS](#)

[IVISPECAN_ATTR_ATTENUATION](#)

[IVISPECAN_ATTR_ATTENUATION_AUTO](#)

[IVISPECAN_ATTR_DETECTOR_TYPE](#)

[IVISPECAN_ATTR_DETECTOR_TYPE_AUTO](#)

[IVISPECAN_ATTR_FREQUENCY_START](#)

[IVISPECAN_ATTR_FREQUENCY_STOP](#)

[IVISPECAN_ATTR_FREQUENCY_OFFSET](#)

[IVISPECAN_ATTR_INPUT_IMPEDANCE](#)

[IVISPECAN_ATTR_NUMBER_OF_SWEEPS](#)

[IVISPECAN_ATTR_REFERENCE_LEVEL](#)

[IVISPECAN_ATTR_REFERENCE_LEVEL_OFFSET](#)

[IVISPECAN_ATTR_RESOLUTION_BANDWIDTH](#)

[IVISPECAN_ATTR_RESOLUTION_BANDWIDTH_AUTO](#)

[IVISPECAN_ATTR_SWEEP_MODE_CONTINUOUS](#)

[IVISPECAN_ATTR_SWEEP_TIME](#)

[IVISPECAN_ATTR_SWEEP_TIME_AUTO](#)

[IVISPECAN_ATTR_TRACE_COUNT](#)

For an instrument with only one Trace, i.e., the Trace Count attribute is one, the driver may return an empty string.

[IVISPECAN_ATTR_TRACE_SIZE](#)

[IVISPECAN_ATTR_TRACE_TYPE](#)

[IVISPECAN_ATTR_VERTICAL_SCALE](#)

[IVISPECAN_ATTR_VIDEO_BANDWIDTH](#)

Related Topics

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

[IviSpecAnBase behavior model](#)

IviSpecAnDeltaMarker Extension Group

The IviSpecAnDeltaMarker extension group supports analyzers that have delta-marker capabilities. A delta marker has the same properties as a normal marker except that its position and amplitude are relative to a fixed reference point. This fixed reference point is defined when the marker is converted from a normal marker to a delta marker.

IviSpecAnDeltaMarker VIs

[IviSpecAn_MakeMarkerDelta](#)

[IviSpecAn_QueryReferenceMarker](#)

IviSpecAnDeltaMarker Properties

IVISPECAN_ATTR_REFERENCE_MARKER_AMPLITUDE

IVISPECAN_ATTR_REFERENCE_MARKER_POSITION

IviSpecAnDisplay Extension Group

The IviSpecAnDisplay extension group controls the display related properties.

IviSpecAnDisplay Properties

IVISPECAN_ATTR_NUMBER_OF_DIVISIONS

IVISPECAN_ATTR_UNITS_PER_DIVISION

IviSpecAnExternalMixer Extension Group

Many spectrum analyzers have outputs and inputs that allow external equipment to use the IF or mixer signal that the spectrum analyzer uses. In this case, external equipment can be used to mix signals to convert them to measurable frequencies. This allows the use of an analyzer to measure values that are outside of the normal frequency range of the equipment. When using an external mixer, many of the settings of the analyzer have to be carefully converted to allow you to know what is meant by the values read. Specifically, the frequency, the harmonic number, mixer configuration, and conversion loss must be configured carefully to be able to use the external mixing successfully. The frequency of the input signal can be expressed as a function of the local oscillator (LO) frequency and the selected harmonic of the 1st LO is as follows:

$$f_{in} = n * f_{LO} \pm f_{IF}$$

Where: f_{in} frequency of input signal
 n order of harmonic used for conversion
 f_{LO} frequency of 1st LO
 f_{IF} intermediate frequency

The Harmonic number defines the order n of the harmonic used for conversion. Both even and odd harmonics can be used. The selected harmonic, together with the setting range of the 1st LO, determines the limits of the settable frequency range. The following applies:

Lower frequency limit: $f_{min} = n * f_{LO,min} - f_{IF}$

Upper frequency limit: $f_{max} = n * f_{LO,max} + f_{IF}$

Where: $f_{LO,min}$ lower frequency limit of LO
 $f_{LO,max}$ upper frequency limit of LO

The following sections describe the mixer configuration and the conversion loss table configuration.

Mixer Configuration

The external mixers are typically configured either as two-port or three port mixers. Single-diode mixers generally require a DC voltage which is applied via the LO line. This DC voltage is to be tuned to the minimum conversion loss versus frequency. Some instruments can define a limit for the BIAS current. The two-port mixer connects the LO OUT / IF IN output of the analyzer to the LO/IF port of the external mixer. The diplexer is contained in the analyzer and the IF signal can be tapped from the line which is used to feed the LO signal to the mixer. The signal to be measured is fed to the RF input of the external mixer. On the other hand, the three-port mixer connects the LO OUT / IF IN output of the analyzer to the LO port of the external mixer. The IF IN input of the analyzer is connected to the IF port of the external mixer. The signal to be measured is fed to the RF input of the external mixer.

Conversion Loss

The maximum settable reference level depends on the external mixers conversion loss which is defined by average conversion loss or by using the conversion loss table. For example, if an IF signal with a level of -20 dBm is applied to the LO OUT / IF IN or IF IN input of the spectrum analyzer, full screen level is attained. Consequently, the maximum settable reference level is -20 dBm at a set conversion loss of 0 dB. If a conversion loss > 0 dB is entered, the maximum settable reference level increases in the same proportion. If the maximum possible reference level is set on the analyzer, this level is reduced if a smaller conversion loss is entered. In addition to the dynamic range of the spectrum analyzer the 1 dB compression point of the mixer has to be taken into account. The levels of the input signals lie well below this value to avoid generation of harmonics of these signals in the mixer. These are converted by the LO signals harmonics of higher order and appear in the displayed spectrum. Some instruments allow the definition of conversion loss tables. The Conversion loss table allows the conversion loss of the mixer in the selected band to be taken into account as a function of frequency. Correction values for frequencies between the individual reference values are obtained by interpolation (Linear interpolation). Outside the frequency range covered by the table the conversion loss is assumed to be the same as that for the reference value marking the table limit.

IviSpecAnExternalMixer VIs

[IviSpecAn_ConfigureConversionLossTable](#)

[IviSpecAn_ConfigureConversionLossTableEnabled](#)

[IviSpecAn_ConfigureExternalMixer](#)

[IviSpecAn_ConfigureExternalMixerBias](#)

[IviSpecAn_ConfigureExternalMixerBiasEnabled](#)

[IviSpecAn_ConfigureExternalMixerEnabled](#)

[IviSpecAn_ConfigureExternalMixerNumberOfPorts](#)

IviSpecAnExternalMixer Properties

IVISPECAN_ATTR_EXTERNAL_MIXER_AVERAGE_CONVERSION_LOSS

IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS

IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS_ENABLED

IVISPECAN_ATTR_EXTERNAL_MIXER_BIAS_LIMIT

IVISPECAN_ATTR_EXTERNAL_MIXER_CONVERSION_LOSS_TABLE_ENTRIES

IVISPECAN_ATTR_EXTERNAL_MIXER_ENABLED

IVISPECAN_ATTR_EXTERNAL_MIXER_HARMONIC

IVISPECAN_ATTR_EXTERNAL_MIXER_NUMBER_OF_PORTS

IviSpecAnExternalTrigger Extension Group

The IviSpecAnExternalTrigger extension group extension group specifies the external trigger level and external trigger slope when the Trigger Source property is set to external, which causes the analyzer to leave the Wait-For-Trigger state.

IviSpecAnTrigger VI

[IviSpecAn_ConfigureExternalTrigger](#)

IviSpecAnTrigger Properties

IVISPECAN_ATTR_EXTERNAL_TRIGGER_LEVEL

IVISPECAN_ATTR_EXTERNAL_TRIGGER_SLOPE

IviSpecAnMarker Extension Group

The IviSpecAnMarker extension group supports spectrum analyzers that have markers. Markers are applied to traces and used for a wide range of operations. Some operations are simple, such as reading an amplitude value at an X-axis position, while others operations are complex, such as signal tracking.

IviSpecAnMarker VIs

[IviSpecAn_ConfigureMarkerEnabled](#)

[IviSpecAn_ConfigureMarkerFrequencyCounter](#)

[IviSpecAn_ConfigureMarkerSearch](#)

[IviSpecAn_ConfigureSignalTrackEnabled](#)

[IviSpecAn_DisableAllMarkers](#)

[IviSpecAn_GetMarkerName](#)

[IviSpecAn_MarkerSearch](#)

[IviSpecAn_MoveMarker](#)

[IviSpecAn_QueryMarker](#)

[IviSpecAn_SetActiveMarker](#)

[IviSpecAn_SetInstrumentFromMarker](#)

IviSpecAnMarker Properties

IVISPECAN_ATTR_ACTIVE_MARKER

IVISPECAN_ATTR_MARKER_AMPLITUDE

IVISPECAN_ATTR_MARKER_COUNT

IVISPECAN_ATTR_MARKER_ENABLED

IVISPECAN_ATTR_MARKER_FREQUENCY_COUNTER_ENABLED

IVISPECAN_ATTR_MARKER_FREQUENCY_COUNTER_RESOLUTION

IVISPECAN_ATTR_MARKER_POSITION

IVISPECAN_ATTR_MARKER_THRESHOLD

IVISPECAN_ATTR_MARKER_TRACE

IVISPECAN_ATTR_PEAK_EXCURSION

IVISPECAN_ATTR_SIGNAL_TRACK_ENABLED

IviSpecAnMarkerType Extension Group

The IviSpecAnMarkerType extension group supports analyzers that have multiple marker types.

IviSpecAnMarkerType VI

[IviSpecAn_QueryMarkerType](#)

IviSpecAnMarkerType Property

IVISPECAN_ATTR_MARKER_TYPE

IviSpecAnMultitrace Extension Group

The IviSpecAnMultitrace extension group defines extensions for analyzers capable of performing simple mathematical VIs on one or more traces.

IviSpecAnMultitrace VIs

[IviSpecAn_AddTraces](#)

[IviSpecAn_CopyTrace](#)

[IviSpecAn_ExchangeTraces](#)

[IviSpecAn_SubtractTraces](#)

IviSpecAnPreselector Extension Group

The IviSpecAnPreselector extension group supports analyzers that have multiple marker types.

IviSpecAnPreselector VI

[IviSpecAn_PeakPreselector](#)

IviSpecAnSoftwareTrigger Extension Group

The IviSpecAnSoftwareTrigger extension group supports spectrum analyzers that can acquire traces based on a software trigger signal. You can send a software trigger to cause signal output to occur.

The IviSpecAnSoftwareTrigger extension group affects instrument behavior when the Trigger Source property is set to Software.

IviSpecAnSoftwareTrigger VI

[IviSpecAn_SendSoftwareTrigger](#)

IviSpecAnTrigger Extension Group

The IviSpecAnTrigger extension group specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state.

IviSpecAnTrigger VI

[IviSpecAn_ConfigureTriggerSource](#)

IviSpecAnTrigger Property

IVISPECAN_ATTR_TRIGGER_SOURCE

IviSpecAnVideoTrigger Extension Group

The IviSpecAnVideoTrigger extension group specifies the video trigger level and video trigger slope when the Trigger Source property is set to Video, which causes the analyzer to leave the Wait-For-Trigger state.

IviSpecAnVideoTrigger VI

[IviSpecAn_ConfigureVideoTrigger](#)

IviSpecAnVideoTrigger Properties

IVISPECAN_ATTR_VIDEO_TRIGGER_LEVEL

IVISPECAN_ATTR_VIDEO_TRIGGER_SLOPE

IviSpecAn Behavior Models

[IviSpecAnBase](#)

[IviSpecAnDeltaMarker](#)

[IviSpecAnDisplay](#)

[IviSpecAnExternalMixer](#)

[IviSpecAnExternalTrigger](#)

[IviSpecAnMarker](#)

[IviSpecAnMultitrace](#)

[IviSpecAnPreselector](#)

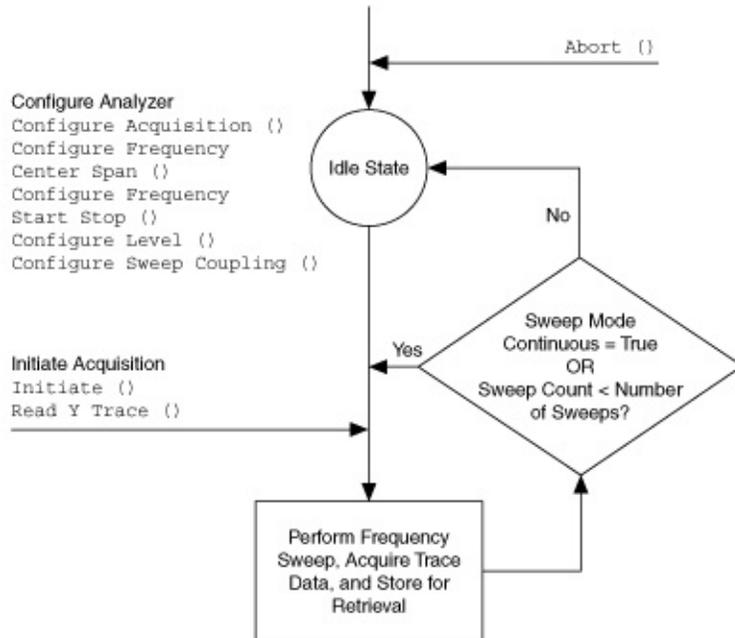
[IviSpecAnSoftwareTrigger](#)

[IviSpecAnTrigger](#)

[IviSpecAnVideoTrigger](#)

IviSpecAn Behavior Model

The following state diagram shows relationships between IviSpecAn Fundamental Capabilities and analyzer behavior.



The main state in the IviSpecAn Class is the Idle state. The analyzer enters the Idle state as the result of being "powered-on", successfully completing a measurement, or by you aborting a previous measurement by you with the [Abort VI](#). Typically, you configure the analyzer while it is in the Idle state. IviSpecAn properties can be configured individually with the Set Attribute VI (IVI-C) or using one of the higher-level VIs.

The [Read Y Trace](#) and [Initiate](#) VIs cause the analyzer to leave the Idle state. The [Read Y Trace](#) VI does not return until the measurement process is complete and the analyzer has returned to the Idle state. The [Initiate](#) VI returns as soon as the analyzer leaves the Idle state.

After the sweep is taken, the analyzer returns to the Idle state. You can use the [Acquisition Status](#) VI to determine if the acquisition is complete or is still in progress.

The [Fetch Y Trace](#) IviSpecAn_Fetch_Y_Trace.html is used to return a waveform from a previously initiated measurement.

IviSpecAnDeltaMarker Behavior Model

The IviSpecAnDeltaMarker extension group follows the same behavior model as the IviSpecAnMarker capability group

IviSpecAnDisplay Behavior Model

The IviSpecAnDisplay extension group follows the same behavior model as the IviSpecAnBase capability group.

IviSpecAnExternalMixer Behavior Model

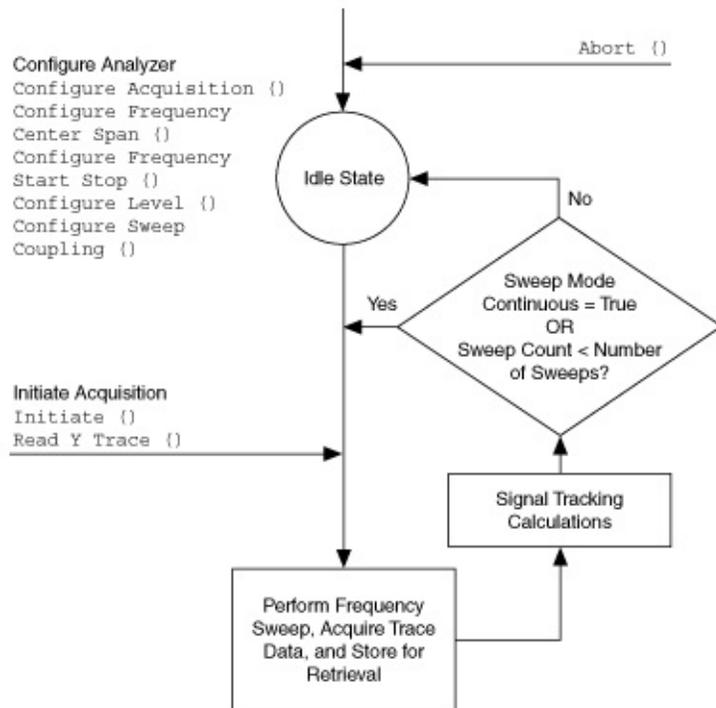
The IviSpecAnExternalMixer extension group follows the same behavior model as the IviSpecAnBase capability group.

IviSpecAnExternalTrigger Behavior Model

The IviSpecAnExternalTrigger extension group follows the same behavior model as the IviSpecAnTrigger extension group

IviSpecAnMarker Behavior Model

The marker properties may be used at any time during the course of the operation of the spectrum analyzer. When signal tracking is enabled, the effects are the same as calling the [Set Instrument From Marker VI](#) with the InstrumentSetting parameter set to FrequencyCenter: it keeps the signal peak at the center of the trace. The adjustment of properties necessary to center the peak is done in the calculation block.



IviSpecAnMarkerType Behavior Model

The IviSpecAnMarkerType extension group follows the same behavior model as the IviSpecAnMarker capability group.

IviSpecAnMultitrace Behavior Model

The IviSpecAnMultitrace extension group follows the same behavior model as the IviSpecAnBase capability group.

IviSpecAnPreselector Behavior Model

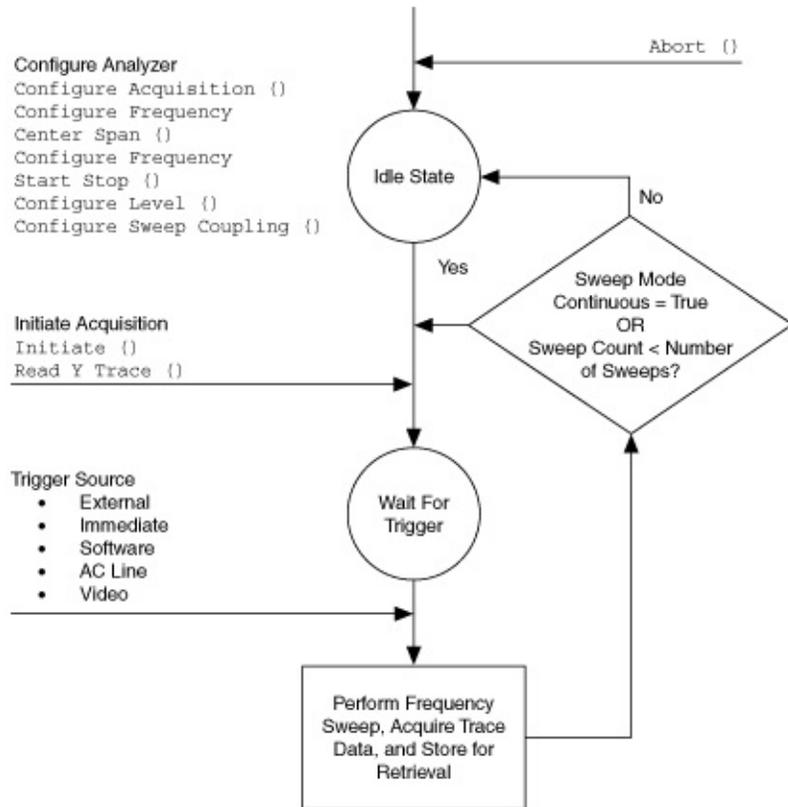
The IviSpecAnPreselector extension group follows the same behavior model as the IviSpecAnBase capability group.

IviSpecAnSoftwareTrigger Behavior Model

The IviSpecAnSoftwareTrigger extension group follows the same behavior model as the [IviSpecAnTrigger](#) extension group.

IviSpecAnTrigger Behavior Model

The IviSpecAnTrigger extension group follows the same behavior model as the [IviSpecAnBase](#) capability group, with the addition of a Wait For Trigger state, as shown below.



IviSpecAnVideoTrigger Behavior Model

The IviSpecAnVideoTrigger extension group follows the same behavior model as the IviSpecAnTrigger extension group .

IviSpecAn VI Tree

Class/Panel Name	VI Name
Configuration Functions	
Configure Frequency Start Stop	<u>IviSpecAn_Configure Frequency Start Stop.vi</u>
Configure Frequency Center Span	<u>IviSpecAn_Configure Frequency Center Span.vi</u>
Configure Frequency Offset	<u>IviSpecAn_Configure Frequency Offset.vi</u>
Configure Sweep Coupling	<u>IviSpecAn_Configure Sweep Coupling.vi</u>
Configure Acquisition	<u>IviSpecAn_Configure Acquisition.vi</u>
Configure Level	<u>IviSpecAn_Configure Level.vi</u>
Configure Trace Type	<u>IviSpecAn_Configure Trace Type.vi</u>
Get Trace Name	<u>IviSpecAn_Get Trace Name.vi</u>
Query Trace Size	<u>IviSpecAn_Query Trace Size.vi</u>
Peak Preselector [PS]	<u>IviSpecAn_Peak Preselector [PS].vi</u>
Multitrace	
Add Traces [MT]	<u>IviSpecAn_Add Traces [MT].vi</u>
Exchange Traces [MT]	<u>IviSpecAn_Exchange Traces [MT].vi</u>
Copy Trace [MT]	<u>IviSpecAn_Copy Trace [MT].vi</u>
Subtract Traces [MT]	<u>IviSpecAn_Subtract</u>

Marker

Configure Marker Enabled [MKR]

[Traces \[MT\].vi](#)

[IviSpecAn_Configure
Marker Enabled
\[MKR\].vi](#)

Configure Marker Frequency Counter [MKR]

[IviSpecAn_Configure
Marker Frequency
Counter \[MKR\].vi](#)

Configure Marker Search [MKR]

[IviSpecAn_Configure
Marker Search
\[MKR\].vi](#)

Configure Signal Track Enabled [MKR]

[IviSpecAn_Configure
Signal Track Enabled
\[MKR\].vi](#)

Disable All Markers [MKR]

[IviSpecAn_Disable All
Markers \[MKR\].vi](#)

Marker Search [MKR]

[IviSpecAn_Marker
Search \[MKR\].vi](#)

Move Marker [MKR]

[IviSpecAn_Move
Marker \[MKR\].vi](#)

Query Marker [MKR]

[IviSpecAn_Query
Marker \[MKR\].vi](#)

Set Active Marker [MKR]

[IviSpecAn_Set Active
Marker \[MKR\].vi](#)

Set Instrument From Marker [MKR]

[IviSpecAn_Set
Instrument From
Marker \[MKR\].vi](#)

Query Marker Type [MKT]

[IviSpecAn_Query
Marker Type \[MKT\].vi](#)

Make Marker Delta [DMK]

[IviSpecAn_Make
Marker Delta \[DMK\].vi](#)

Query Reference Marker [DMK]

[IviSpecAn_Query
Reference Marker
\[DMK\].vi](#)

Get Marker Name [MKR]

[IviSpecAn_Get Marker Name \[MKR\].vi](#)

Trigger Extension Group

Configure Trigger Source [TRG]

[IviSpecAn_Configure Trigger Source \[TRG\].vi](#)

Configure External Trigger [EXT]

[IviSpecAn_Configure External Trigger \[EXT\].vi](#)

Configure Video Trigger [VT]

[IviSpecAn_Configure Video Trigger \[VT\].vi](#)

External Mixer Extension Group

Configure External Mixer Enabled [EM]

[IviSpecAn_Configure External Mixer Enabled \[EM\].vi](#)

Configure External Mixer Number of Ports [EM]

[IviSpecAn_Configure External Mixer Number of Ports \[EM\].vi](#)

Configure External Mixer [EM]

[IviSpecAn_Configure External Mixer \[EM\].vi](#)

Configure External Mixer Bias Enabled [EM]

[IviSpecAn_Configure External Mixer Bias Enabled \[EM\].vi](#)

Configure External Mixer Bias [EM]

[IviSpecAn_Configure External Mixer Bias \[EM\].vi](#)

Configure Conversion Loss Table Enabled [EM]

[IviSpecAn_Configure Conversion Loss Table Enabled \[EM\].vi](#)

Configure Conversion Loss Table [EM]

[IviSpecAn_Configure Conversion Loss Table \[EM\].vi](#)

Measurement

Read Y Trace

[IviSpecAn_Read Y](#)

Fetch Y Trace

[Trace.vi](#)

[IviSpecAn_Fetch Y
Trace.vi](#)

Low Level Measurement

Initiate

[IviSpecAn_Initiate.vi](#)

Abort

[IviSpecAn_Abort.vi](#)

Acquisition Status

[IviSpecAn_Acquisition
Status.vi](#)

Send Software Trigger [SWT]

[IviSpecAn_Send
Software Trigger
\[SWT\].vi](#)

IviSpecAn Properties

Group/Attribute Name	Property Link
Basic Operation	
Amplitude Units	<u>IviSpecAn»Basic Operation»Amplitud</u>
Attenuation	<u>IviSpecAn»Basic Operation»Attenuati</u>
Attenuation Auto	<u>IviSpecAn»Basic Operation»Attenuati</u>
Detector Type	<u>IviSpecAn»Basic Operation»Detector</u>
Detector Type Auto	<u>IviSpecAn»Basic Operation»Detector Auto</u>
Frequency Start	<u>IviSpecAn»Basic Operation»Frequen</u>
Frequency Stop	<u>IviSpecAn»Basic Operation»Frequen</u>
Frequency Offset	<u>IviSpecAn»Basic Operation»Frequen</u>
Input Impedance	<u>IviSpecAn»Basic Operation»Input Imp</u>
Number of Sweeps	<u>IviSpecAn»Basic Operation»Number Sweeps</u>
Reference Level	<u>IviSpecAn»Basic Operation»Referenc</u>
Reference Level Offset	<u>IviSpecAn»Basic Operation»Referenc Offset</u>
Resolution Bandwidth	<u>IviSpecAn»Basic Operation»Resolutio Bandwidth</u>

Resolution Bandwidth Auto

[IviSpecAn»Basic
Operation»Resolutio
Bandwidth Auto](#)

Sweep Mode Continuous

[IviSpecAn»Basic
Operation»Sweep M
Continuous](#)

Sweep Time

[IviSpecAn»Basic
Operation»Sweep T](#)

Sweep Time Auto

[IviSpecAn»Basic
Operation»Sweep T](#)

Trace Count

[IviSpecAn»Basic
Operation»Trace Cc](#)

Trace Size

[IviSpecAn»Basic
Operation»Trace Si](#)

Trace Type

[IviSpecAn»Basic
Operation»Trace Ty](#)

Vertical Scale

[IviSpecAn»Basic
Operation»Vertical S](#)

Video Bandwidth

[IviSpecAn»Basic
Operation»Video Ba](#)

Video Bandwidth Auto

[IviSpecAn»Basic
Operation»Video Ba
Auto](#)

Markers

Active Marker [MKR]

[IviSpecAn»Markers:
Marker \[MKR\]](#)

Marker Amplitude [MKR]

[IviSpecAn»Markers:
Amplitude \[MKR\]](#)

Marker Enabled [MKR]

[IviSpecAn»Markers:
Enabled \[MKR\]](#)

Marker Frequency Counter Enabled [MKR]

[IviSpecAn»Markers:
Frequency Counter
\[MKR\]](#)

Marker Frequency Counter Resolution [MKR]

[IviSpecAn»Markers: Frequency Counter Resolution \[MKR\]](#)

Marker Position [MKR]

[IviSpecAn»Markers: Position \[MKR\]](#)

Marker Threshold [MKR]

[IviSpecAn»Markers: Threshold \[MKR\]](#)

Marker Trace [MKR]

[IviSpecAn»Markers: Trace \[MKR\]](#)

Number of Markers [MKR]

[IviSpecAn»Markers: of Markers \[MKR\]](#)

Peak Excursion [MKR]

[IviSpecAn»Markers: Excursion \[MKR\]](#)

Signal Track Enabled [MKR]

[IviSpecAn»Markers: Track Enabled \[MKR\]](#)

Marker Type

Marker Type [MKT]

[IviSpecAn»Markers: Type»Marker Type \[MKT\]](#)

Delta Marker

Reference Marker Amplitude [DMK]

[IviSpecAn»Markers: Marker»Reference Amplitude \[DMK\]](#)

Reference Marker Position [DMK]

[IviSpecAn»Markers: Marker»Reference Position \[DMK\]](#)

Trigger

Trigger Source [TRG]

[IviSpecAn»Trigger»Source \[TRG\]](#)

External Trigger

External Trigger Level [EXT]

[IviSpecAn»Trigger»Trigger»External Trigger Level \[EXT\]](#)

External Trigger Slope [EXT]

[IviSpecAn»Trigger»Slope \[EXT\]](#)

[Trigger»External Tri
Slope \[EXT\]](#)

Video Trigger

Video Trigger Level [VT]

[IviSpecAn»Trigger»
Trigger»Video Trigg
\[VT\]](#)

Video Trigger Slope [VT]

[IviSpecAn»Trigger»
Trigger»Video Trigg
\[VT\]](#)

Display Control

Units Per Division [DSP]

[IviSpecAn»Display
Control»Units Per D
\[DSP\]](#)

Number of Divisions [DSP]

[IviSpecAn»Display
Control»Number of
\[DSP\]](#)

External Mixing

External Mixer Enabled [EM]

[IviSpecAn»External
Mixing»External Mix
Enabled \[EM\]](#)

External Mixer Number of Ports [EM]

[IviSpecAn»External
Mixing»External Mix
Number of Ports \[EM\]](#)

External Mixer Harmonic [EM]

[IviSpecAn»External
Mixing»External Mix
Harmonic \[EM\]](#)

External Mixer Average Conversion Loss [EM]

[IviSpecAn»External
Mixing»External Mix
Average Conversior
\[EM\]](#)

External Mixer Conversion Loss Table Enabled [EM]

[IviSpecAn»External
Mixing»External Mix
Conversion Loss Ta
Enabled \[EM\]](#)

Bias Enabled [EM]

[IviSpecAn»External
Mixing»Bias Enable](#)

Bias [EM]

[IviSpecAn»External
Mixing»Bias \[EM\]](#)

Bias Limit [EM]

[IviSpecAn»External
Mixing»Bias Limit \[E](#)

IviSpecAn Error and Warning Codes

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviSpecAn Errors	0xBFFFA2001 to 0xBFFFA3FFF
IviSpecAn Warnings	0x3FFFA2001 to 0x3FFFA3FFF
IVI Specific Driver Errors	0xBFFFA4000 to 0xBFFFA5FFF
IVI Specific Driver Warnings	0x3FFFA4000 to 0x3FFFA5FFF
IVI Errors	0xBFFFA0000 to 0xBFFFA1FFF
IVI Warnings	0x3FFFA0000 to 0x3FFFA1FFF
Common Instrument Driver Errors	0xBFFFC0000 to 0xBFFFCFFFF
Common Instrument Driver Warnings	0x3FFFC0000 to 0x3FFFCFFFF
VISA Errors	0xBFFF0000 to 0xBFFFFFFF
VISA Warnings	0x3FFF0000 to 0xFFFFFFF

The IviSpecAn class driver defines the error codes shown in the following table in addition to the IVI defined error codes.

IviSpecAn Error Codes

Error	Value	Message
IVISPECAN_ERROR_MARKER_NOT_ENABLED	0xBFFA2001	The Active Marker is not enabled.
IVISPECAN_ERROR_NOT_DELTA_MARKER	0xBFFA2002	The Active Marker is not a delta marker.
IVISPECAN_ERROR_MAX_TIME_EXCEEDED	0xBFFA2003	Maximum time exceeded before the operation completed

IviSpecAn Warning Code

Warning	Value	Message
IVISPECAN_WARN_OVER_RANGE	0x3FFA2002	Over Range warning
IVISPECAN_WARN_MEASURE_UNCALIBERATED	0x3FFA2001	The instrument was in an uncalibrated state when the measurement was taken

Related Topic

[IVI Status Codes](#)

IviSpecAn VIs

Expand this book to view an alphabetized list of IviSpecAn VIs.

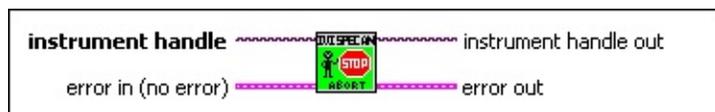
IviSpecAn Abort

IviSpecAnBase Capability Group

This VI aborts a previously initiated measurement and returns the spectrum analyzer to the idle state.

Note:

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. Use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.



instrument handle The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

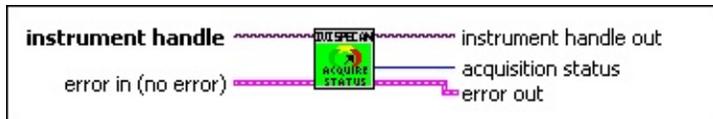
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Acquisition Status

IviSpecAnBase Capability Group

This VI determines if an acquisition is in progress or complete.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

E/W **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **acquisition status** Returns the acquisition status.

Valid Values:

IVISPECAN_VAL_ACQUISITION_STATUS_COMPLETE (1) - The spectrum analyzer has completed the acquisition.

IVISPECAN_VAL_ACQUISITION_STATUS_IN_PROGRESS (0) - The spectrum analyzer is still acquiring data.

IVISPECAN_VAL_ACQUISITION_STATUS_UNKNOWN (-1) - The spectrum analyzer cannot determine the status of the acquisition.

E/W **error out** The error out cluster passes error or warning information

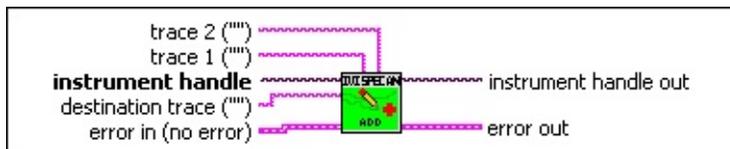
out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Add Traces [MT]

IviSpecAnMultitrace Capability Group

This VI adds trace 1 and trace 2, point by point, and stores the results in the destination trace. Any data in the destination trace is deleted.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **destination trace** Pass the trace name into which the added traces are stored. Any data in the destination trace is overwritten.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

abc **trace 1** Pass the trace name of one of the two traces to be added.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to

change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

 **trace 2** Pass the trace name of the second trace to be added.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

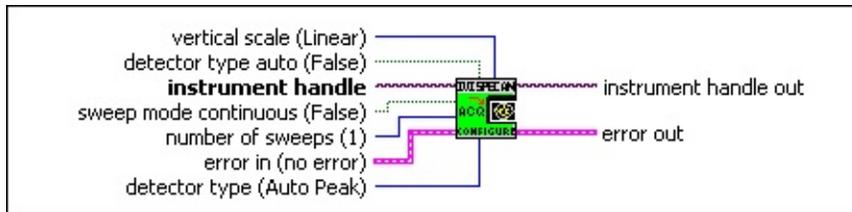
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Acquisition

IviSpecAnBase Capability Group

This VI configures the acquisition properties of the spectrum analyzer.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **sweep mode continuous** Turns on/off continuous sweeping. The driver uses this value to set the property.

Valid Values:

TRUE - Sweep mode is continuous.

FALSE - Sweep mode is not continuous.

Default Value: FALSE

I32 **number of sweeps** The number of sweeps to take. The driver sets the property to this value.

Default Value: 1

TF **detector type auto** Enables the auto detector. The driver uses this value to set the `_AUTO` property.

Note:

When the `_AUTO` property is set to True, the relationship between the property and the `IVISPECAN__ATTR_DETECTOR_TYP` attribute is not defined.

Valid Values:

TRUE - Detector type is automatically selected.
FALSE - Detector type is manually selected.

Default Value: FALSE

I32 **detector type** Specifies the method of capturing and processing signal data. The driver uses this value to set the property.

Valid Values:

- IVISPECAN_VAL_DETECTOR_TYPE_AUTO_PEAK - Allows the detector to capture better readings by using both positive and negative peak values when noise is present.

- IVISPECAN_VAL_DETECTOR_TYPE_AVERAGE - Average value of samples taken within the bin for a dedicated point on the display.

- IVISPECAN_VAL_DETECTOR_TYPE_MAX_PEAK - Obtains the maximum video signal between the last display point and the present display point.

- IVISPECAN_VAL_DETECTOR_TYPE_MIN_PEAK - Obtains the minimum video signal between the last display point and the present display point.

- IVISPECAN_VAL_DETECTOR_TYPE_SAMPLE - Pick one point within a bin.

- IVISPECAN_VAL_DETECTOR_TYPE_RMS - RMS value of samples taken within the bin for a dedicated point on the display.

Default Values: IVISPECAN_VAL_DETECTOR_TYPE_AVERAGE

I32 **vertical scale** Pass the vertical scale of the measurement hardware that you want the spectrum analyzer to use (use of log amplifiers or linear amplifiers). The driver sets the property to this value.

Defined Values:

IVISPECAN_VAL_VERTICAL_SCALE_LINEAR - Linear
IVISPECAN_VAL_VERTICAL_SCALE_LOGARITHMIC -
Logarithmic

Default Value:

IVISPECAN_VAL_VERTICAL_SCALE_LOGARITHMIC

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

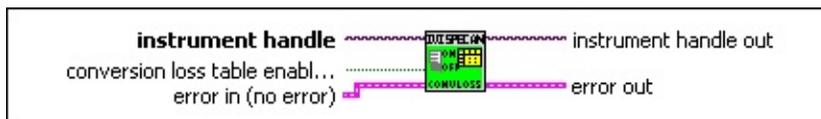
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Conversion Loss Table Enabled [EM]

IviSpecAnExternalMixer Capability Group

This VI enables the conversion loss table.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **conversion loss table enabled** Specify whether the conversion loss table is enabled. The driver uses this value to set the IviSpecAn>>External Mixing>>External Mixer Conversion Loss Table Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

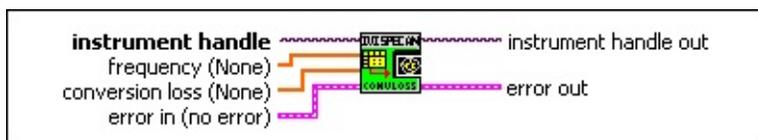
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Conversion Loss Table [EM]

IviSpecAnExternalMixer Capability Group

This VI configures the conversion loss table by specifying a series of frequency and a power loss pairs.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **count** Specify the number of frequency and conversion loss pairs.

Default Value: None

abc **frequency** An array of frequency values for the frequency and power loss pairs. The array must have at least as many elements as the value you specify in the Count parameter.

Default Value: None

abc **conversion loss** An array of conversion loss values for the frequency and power loss pairs. The array must have at least as many elements as the value you specify in the Count parameter.

Default Value: None

err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#)

VIs.



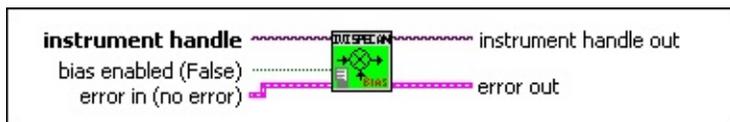
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Mixer Bias Enabled [EM]

IviSpecAnExternalMixer Capability Group

This VI enables the external mixing bias.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **bias enabled** Specifies whether external mixer bias is enabled. The driver uses this value to set the IviSpecAn>>External Mixing>>Bias Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

E+ **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

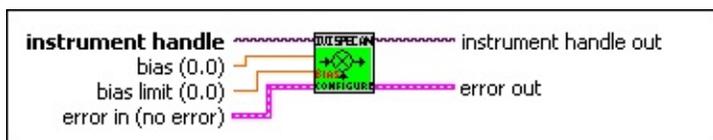
E+ **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Mixer Bias [EM]

IviSpecAnExternalMixer Capability Group

This VI configures the external mixer bias and the external mixer bias limit.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **bias** Specifies the bias current. The driver uses this value to set the IviSpecAn>>External Mixing>>Bias property.

Units: Amps

Default Value: 0.0

DBL **bias limit** Specifies the bias current limit. The driver uses this value to set the IviSpecAn>>External Mixing>>Bias_LIMIT property.

Units: Amps

Default Value: 0.0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#)

VIs.



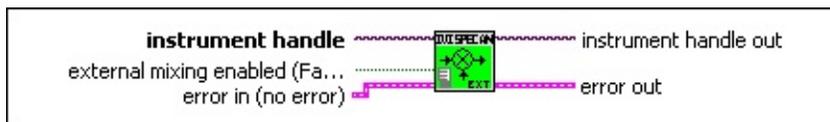
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Mixer Enabled [EM]

IviSpecAnExternalMixer Capability Group

This VI enables external mixing.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **external mixing enabled** Specifies whether external mixing is enabled. The driver uses this value to set the IviSpecAn>>External Mixing>>External Mixer Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

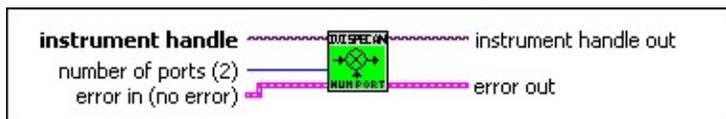
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Mixer Number of Ports [EM]

IviSpecAnExternalMixer Capability Group

This VI specifies the number of external mixer ports.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **number of ports** Specify the number of mixer ports. The driver sets the IviSpecAn>>External Mixing>>External Mixer Number of Ports property to this value.

Default Value: 2

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

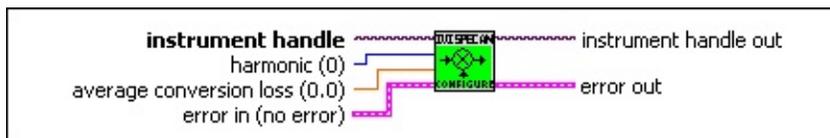
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Mixer [EM]

IviSpecAnExternalMixer Capability Group

This VI specifies the mixer harmonic and average conversion loss.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **harmonic** Specifies the order n of the harmonic used for conversion. The driver sets the IviSpecAn>>External Mixing>>External Mixer Harmonic property to this value.

Default Value: 0

DBL **average conversion loss** Specifies the average conversion loss. The driver sets the IviSpecAn>>External Mixing>>External Mixer Average Conversion Loss to this value.

Default Value: 0.0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

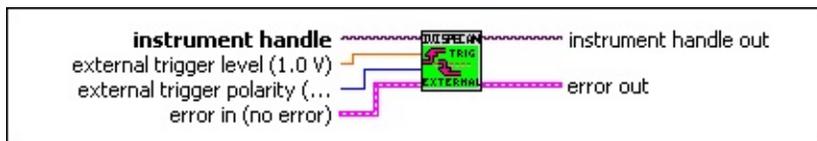
Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure External Trigger [EXT]

IviSpecAnExternalTrigger Capability Group

This VI specifies the external level and polarity for triggering. This is applicable when the trigger source is set to external.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **external trigger level** Specifies the level of the external trigger signal to trigger an acquisition. The driver uses this value to set the IviSpecAn>>Trigger>>External Trigger>>External Trigger Level property.

Units: Volts

Default Value: 1.0 V

I32 **external trigger polarity** Specifies the slope of the external trigger signal to trigger an acquisition. The driver uses this value to set the IviSpecAn>>Trigger>>External Trigger>>External Trigger Slope property.

Valid Values:

- IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_POSITIVE -
Sets positive slope.

- IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_NEGATIVE -
Sets positive slope.

Default value:

IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_POSITIVE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Frequency Center Span

IviSpecAnBase Capability Group

This VI configures the frequency range of the spectrum analyzer using the center frequency and the frequency span. If span corresponds to zero Hertz, then the spectrum analyzer operates in time-domain mode. Otherwise, spectrum analyzer operates in frequency-domain mode.

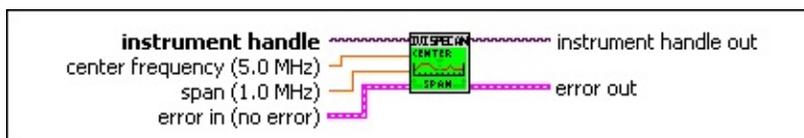
This VI modifies the and properties as follows:

Frequency Start = Center Frequency - Span/2

Frequency Stop = Center Frequency + Span/2

Note:

In auto-coupled mode, resolution bandwidth (RBW), video bandwidth (VBW), and sweep time may be affected by this VI.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **center frequency** The center frequency of the frequency sweep.

Units: Hertz

Default Value: 5.0 MHz

DBL **span** The frequency span of the frequency sweep.

Units: Hertz

Default Value: 1.0 MHz



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



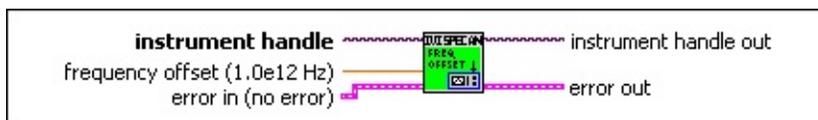
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Frequency Offset

IviSpecAnBase Capability Group

This VI configures the frequency offset of the spectrum analyzer. This affects the setting of the spectrum analyzer's absolute frequencies, such as start, stop, center, and marker. This VI does not modify the settings for differences of frequencies, such as span and delta marker.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **frequency offset** Specifies an offset value that is added to the frequency readout. The offset is used to compensate for external frequency conversion. The driver uses this value to set the property.

Units: Hertz

Default Value: 1.0e12 Hz

ERR **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

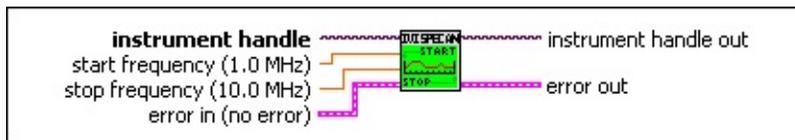
ERR **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Frequency Start Stop

IviSpecAnBase Capability Group

This VI configures the frequency range defining its start frequency and its stop frequency. If the start frequency is equal to the stop frequency, then the spectrum analyzer operates in the time-domain mode. Otherwise, the spectrum analyzer operates in frequency-domain mode.



Notes

(1) In auto-coupled mode, resolution bandwidth (RBW), video bandwidth (VBW), and sweep time may be affected by this VI.

 **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **start frequency** The start frequency of the frequency sweep. The driver uses this value to set the property.

Units: Hertz

Default Value: 1.0 MHz

 **stop frequency** The stop frequency of the frequency sweep. The driver uses this value to set the property.

Units: Hertz

Default Value: 10.0 MHz

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if

any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



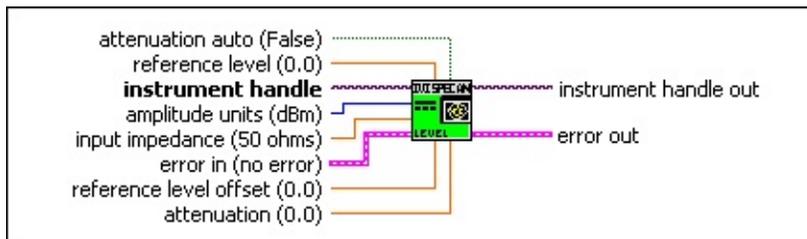
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Level

IviSpecAnBase Capability Group

This VI configures the vertical settings of the spectrum analyzer. This corresponds to settings like amplitude units, input attenuation, input impedance, reference level, and reference level offset.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **amplitude units** Pass the units you want the spectrum analyzer to use for input, output, and display. The driver sets the to this value.

Valid Values:

- IVISPECAN_VAL_AMPLITUDE_UNITS_DBM - Sets the spectrum analyzer to measure in decibels relative to 1 milliwatt.
- IVISPECAN_VAL_AMPLITUDE_UNITS_DBMV - Sets the spectrum analyzer to measure in decibels relative to 1 millivolt.
- IVISPECAN_VAL_AMPLITUDE_UNITS_DBUV - Sets the spectrum analyzer to measure in decibels relative to 1 microvolt.
- IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT - Sets the spectrum analyzer to measure in volts.
- IVISPECAN_VAL_AMPLITUDE_UNITS_WATT - Sets the spectrum analyzer to measure in watts.

Default value: IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT

 **input impedance** Specifies the value of input impedance, in ohms, expected at the active input port. This is typically 50 ohms or 75 ohms. The driver uses this value to set the property.

Units: Ohms

Default Value: 50 ohms

 **reference level** The calibrated vertical position of the captured data used as a reference for amplitude measurements. This is typically set to a value slightly higher than the highest expected signal level. The driver uses this value to set the property.

Units: The units are determined by setting the Amplitude Units parameter of this VI or by setting property.

Default Value: 0.0

 **reference level offset** Specifies the offset value to the reference level. This adjusts the reference level for external signal gain or loss. A positive value corresponds to a gain while a negative number corresponds to a loss. The driver uses this value to set the `_OFFSET` property.

Units: dB

Default Value: 0.0

 **attenuation auto** Specifies whether the spectrum analyzer is enabled for auto attenuation. When you set this value to TRUE, the attenuation is automatically selected. When you set this value to FALSE, the attenuation is set to manual selection. The driver uses this value to set the `_AUTO` property.

Valid Values:

TRUE - Attenuation is automatically selected

FALSE - Attenuation is manually selected

Default Value: FALSE



attenuation Specifies the input attenuation (in positive dB). This means that if 10dB is selected, the result is a reduction in the signal level of 10 dB. The driver uses this value to set the property.

Units: dB

Default Value: 0.0

Note:

If auto attenuation is turned on, this value is ignored.



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



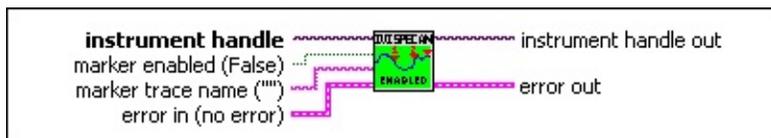
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Marker Enabled [MKR]

IviSpecAnMarker Capability Group

This VI enables the active marker on the specified trace.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **marker enabled** Specifies whether the specified marker is enabled. The driver uses this value to set the IviSpecAn>>Markers>>Marker Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

abc **marker trace name** Pass the name of the trace you for which you are enabling the active marker.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

Err [error in \(no error\)](#) The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



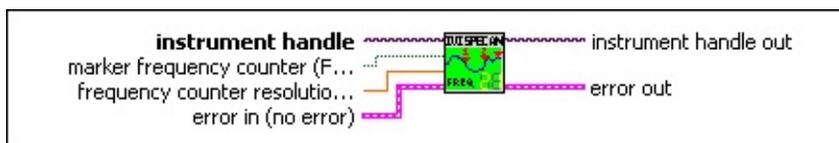
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Marker Frequency Counter [MKR]

IviSpecAnMarker Capability Group

This VI sets the marker frequency counter resolution and turns the marker frequency counter on/off.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **marker frequency counter** Specifies whether the frequency counter is enabled. The driver uses this value to set the IviSpecAn>>Markers>>Marker Frequency Counter Enabled property.

Valid Values: TRUE; FALSE

Default Value: FALSE

DBL **frequency counter resolution** Specifies the frequency counter resolution. The measurement gate time is the reciprocal of the specified resolution. The driver uses this value to set the IviSpecAn>>Markers>>Marker Frequency Counter Resolution property.

Units: Hertz

Default Value: 1.0 KHz

Note:

This value is ignored when the IviSpecAn>>Markers>>Marker

Frequency Counter Enabled is False.



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



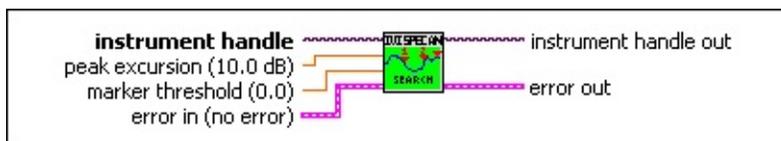
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Marker Search [MKR]

IviSpecAnMarker Capability Group

This VI configures the marker peak excursion and marker threshold values. The marker peak excursion specifies the minimum amplitude variation that can be recognized as a peak or minimum by the marker. The marker threshold specifies a lower bound for ALL marker search VIs.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **peak excursion** Specifies the minimum amplitude variation of the signal in dB that the identifies as a peak. The driver uses this value to set the IviSpecAn>>Markers>>Peak Excursion property.

Units: dB

Default Value: 10.0 dB

DBL **marker threshold** Specifies the lower limit of the search domain vertical range for the . The driver uses this value to set the IviSpecAn>>Markers>>Marker Threshold property.

Units: Amplitude Units

Default Value: 0.0

Err **[error in \(no error\)](#)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Signal Track Enabled [MKR]

IviSpecAnMarker Capability Group

This VI enables signal-tracking. When signal tracking is turned on, the spectrum analyzer centers the signal on which the active marker resides after each sweep. There are two prevalent methods of signal tracking in the industry.

1. A search for the largest signal on screen is made, and the spectrum analyzer center frequency is tuned to this signal.
2. The marker is placed on a signal (anywhere on screen) and when marker track is enabled, the frequency of the peak of the selected signal is used for tuning the spectrum analyzer. This does not need to be the largest signal on screen, and this method tracks the specified signal in the presence of a larger signal.

This VI implements the first method of signal tracking. This INVALIDATES the and property values.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TFI **signal track enabled** Specifies whether signal tracking is enabled. The driver uses this value to set the IviSpecAn>>Markers>>Signal Track Enabled property.

Note: Signal track can only be enabled on one marker at a time. The driver shall check all other markers to see if this VI is already enabled on any marker other than the active and turn this off on the other marker before enabling this on the active marker.

Valid Values: TRUE; FALSE

Default Value: FALSE

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

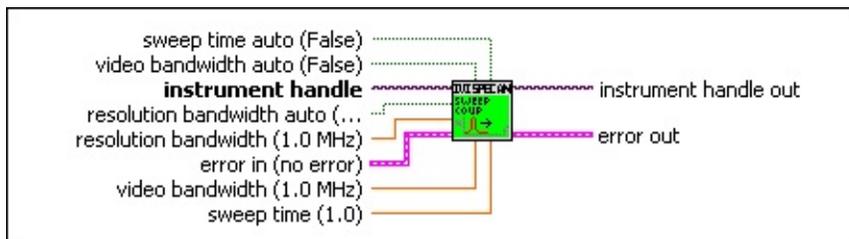
 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Sweep Coupling

IviSpecAnBase Capability Group

This VI configures the coupling and sweeping properties of the spectrum analyzer.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **resolution bandwidth auto** Enables resolution bandwidth auto coupling. The driver uses this value to set the `_AUTO` property.

Valid Values:

TRUE - Resolution bandwidth is automatically selected.

FALSE - Resolution bandwidth is manually selected.

Default Value: FALSE

DBL **resolution bandwidth** Specifies the measurement resolution bandwidth. The driver uses this value to set the property.

Units: Hertz

Default Value: 1.0 HHZ

Note:

This value is ignored when the Resolution Bandwidth Auto parameter or the Auto is set to TRUE.

 **video bandwidth auto** Enables video bandwidth auto coupling. The driver uses this value to set the `_AUTO` property.

Valid Values:

TRUE - Video bandwidth is automatically selected

FALSE - Video bandwidth is manually selected

Default Value: FALSE

 **video bandwidth** Specifies the video bandwidth of the post-detection filter. The driver uses this value to set the property.

Units: Hertz

Default Value: 1.0 MHz

Note:

This value is ignored when the Video Bandwidth Auto parameter or the `_AUTO` is set to TRUE.

 **sweep time auto** Enables sweep time auto. The driver uses this value to set the `_AUTO` property.

Valid Values:

TRUE - Sweep time is automatically selected

FALSE - Sweep time is manually selected

Default Value: FALSE

 **sweep time** The length of time to complete one sweep. The driver uses this value to set the property.

Units: seconds

Default Value: 1.0

Note:

This value is ignored when the Sweep Time Auto parameter or the

_AUTO is set to TRUE.



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



[instrument handle out](#) The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



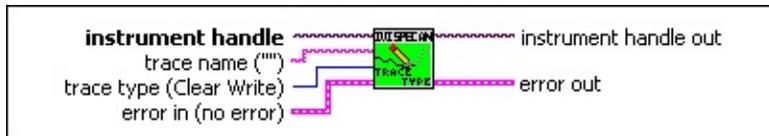
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Trace Type

IviSpecAnBase Capability Group

This VI configures the type of trace to acquire.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.
- abc** **trace name** Pass the name of the trace you are configuring.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

- I32** **trace type** The type of trace to acquire. The driver sets the property to this value.

Valid Values:

IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE
- Sets the spectrum analyzer to clear previous sweep data off the display before performing a sweep. Subsequent sweeps may or may not clear the display first, but the data array at the end of the sweep is entirely new.

IVISPECAN_VAL_TRACE_TYPE_MAX_HOLD

- Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, whichever is higher.

IVISPECAN_VAL_TRACE_TYPE_MIN_HOLD

- Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, whichever is lower.

IVISPECAN_VAL_TRACE_TYPE_VIDEO_AVERAGE

- Sets the spectrum analyzer to maintain a running average of the swept data.

IVISPECAN_VAL_TRACE_TYPE_VIEW

- Disables acquisition into this trace but displays the existing trace data.

IVISPECAN_VAL_TRACE_TYPE_STORE

- Disables acquisition and disables the display of the existing trace data.

Default Value: IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



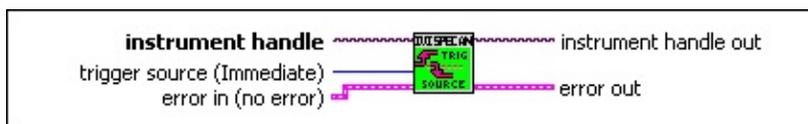
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Trigger Source [TRG]

IviSpecAnTrigger Capability Group

This VI specifies the trigger source that causes the spectrum analyzer to leave the Wait-for-Trigger state.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **trigger source** Specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state. The driver uses this value to set the IviSpecAn>>Trigger>>Trigger Source property.

Valid Values:

- IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNAL - The spectrum analyzer waits until it receives a trigger on the external trigger connector.
- IVISPECAN_VAL_TRIGGER_SOURCE_IMMEDIATE - The spectrum analyzer does not wait for a trigger of any kind.
- IVISPECAN_VAL_TRIGGER_SOURCE_AC_LINE - The spectrum analyzer waits until it receives a trigger on the AC line.
- IVISPECAN_VAL_TRIGGER_SOURCE_VIDEO - The spectrum analyzer waits until it receives a video level.

Default value:

IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNAL



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



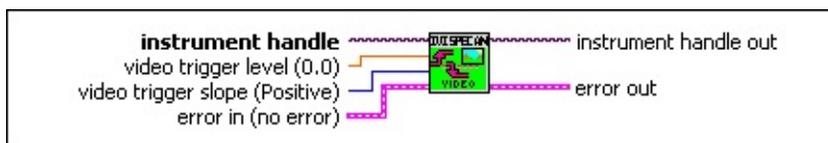
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Configure Video Trigger [VT]

IviSpecAnVideoTrigger Capability Group

This VI specifies the video level and polarity for video triggering. This is applicable when the trigger source is set to video.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **video trigger level** Specifies the level of the video signal to trigger an acquisition. The driver uses this value to set the IviSpecAn>>Trigger>>Video Trigger>>Video Trigger Level property.

Units: The units are specified by the property.

Default Value: 0.0

I32 **video trigger slope** Specifies the slope of the video signal to trigger an acquisition. The driver uses this value to set the IviSpecAn>>Trigger>>Video Trigger>>Video Trigger Slope property.

Valid Values:

- IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POSITIVE - Sets positive slope.
- IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_NEGATIVE - Sets positive slope.

Default value:

IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POSITIVE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



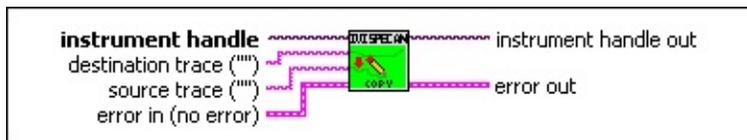
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Copy Trace [MT]

IviSpecAnMultitrace Capability Group

This VI copies one trace array to another trace array. Any data in the destination trace is over written.



- instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.
- destination trace** Pass the trace name into which the array is stored.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

- source trace** Pass the trace name of the source trace array.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to

change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



[instrument handle out](#) The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Disable All Markers [MKR]

IviSpecAnMarker Capability Group

This VI disables all of markers.



 **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

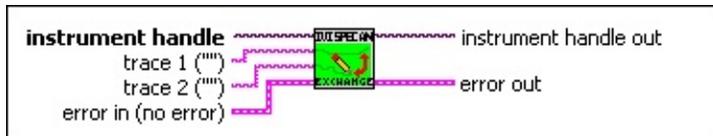
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Exchange Traces [MT]

IviSpecAnMultitrace Capability Group

This VI exchanges the data arrays of two traces.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **trace 1** Pass the trace name of one of the two traces to be exchanged.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

abc **trace 2** Pass the trace name of the second of the two traces to be exchanged.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to

change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Fetch Y Trace

IviSpecAnBase Capability Group

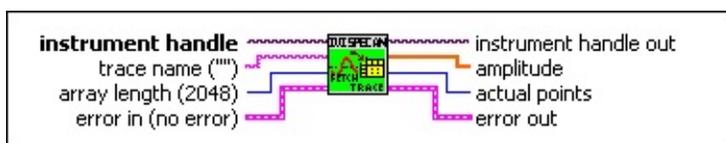
This VI returns the trace the spectrum analyzer acquires. The trace is from a previously initiated acquisition. Call the to start an acquisition. Call the to determine when the acquisition is complete. Once the acquisition is complete, call the Fetch Y Trace VI to return the trace.

You may call the instead of the . The starts an acquisition, waits for the acquisition to complete, and returns the trace in one VI call.

Use the when acquiring multiple traces. Use a different trace name for each call. You may also call after calling to acquire subsequent traces.

Note:

This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **trace name** Pass the name of the trace for the desired amplitude array.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can

differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

 **array length** Pass the number of Amplitude array points requested.

Default Value: None

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **actual points** Indicates the number of points the VI places in the Amplitude array.

Use the to determine the number of points the spectrum analyzer acquires.

 **amplitude** Returns the amplitude waveform that the spectrum analyzer acquires.

This data represents the amplitude of the signals of the sweep from the start frequency to the stop frequency (in frequency domain, in time domain the amplitude array is ordered from beginning of sweep to end).

The units are determined by the Amplitude Units parameter of the or by the property.

Note:

The returns the number of points the spectrum analyzer acquires.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Get Marker Name [MKR]

IviSpecAnMarker Capability Group

This VI returns the specific driver defined trace name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the IviSpecAn>>Markers>>Number of Markers property, the VI returns an empty string in the Name parameter and returns the Invalid Value error.

Note: For an instrument with only one Marker, i.e. the IviSpecAn>>Markers>>Number of Markers property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **index** Specifies the index of the Marker Name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain

from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



name The marker name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

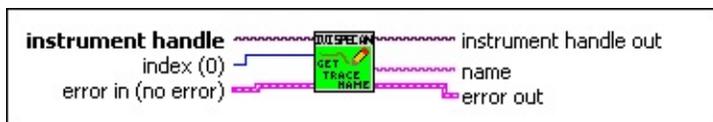
IviSpecAn Get Trace Name

IviSpecAnBase Capability Group

This VI returns the specific driver defined trace name that corresponds to the one-based index specified by the Index parameter. If you pass in a value for the Index parameter that is less than one or greater than the value of the property, the VI returns an empty string in the Name parameter and returns the Invalid Value error (0xBFFA1010).

Note:

For an instrument with only one Trace, i.e. the property is one, the driver may return an empty string.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **index** Specifies the index of the Trace Name to return.

Default Value: 0

I32 **name buffer size** Specify the buffer size for the Name parameter.

Default Value: 0

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **name** The trace name that corresponds to the Index.

The Name buffer must contain at least as many elements as the value you specify with the Name Buffer Size parameter.



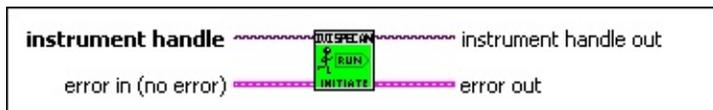
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Initiate

IviSpecAnBase Capability Group

This VI initiates an acquisition. After calling this VI, the spectrum analyzer leaves the idle state and waits for a trigger.



Notes

(1) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

(2) Call to determine when the acquisition is complete.

 **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.

IviSpecAn Make Marker Delta [DMK]

IviSpecAnDeltaMarker Capability Group

This VI specifies whether the active marker is a delta marker.

When this VI is called with DeltaMarker true, the current active marker is changed to a delta marker and the associated reference marker is moved to the current position of the active marker. The current position becomes the reference point for marker values. The marker readout indicates the relative frequency (or time) separation and amplitude difference between the reference and active marker.

When this Delta Marker is set to false, the current marker is changed to a normal marker. The reference marker is determined by calling the .



Notes

(1) If the current active marker is not enabled then this VI enables the active marker.

I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

TF **delta marker** Specifies whether marker Delta is True or False. Depending on this value, the driver will set the marker to `IVISPECAN_VAL_MARKER_TYPE_NORMAL` or `IVISPECAN_VAL_MARKER_TYPE_DELTA` (see the VI help for more details).

Valid Values: TRUE; FALSE

Default Value: FALSE



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



[instrument handle out](#) The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

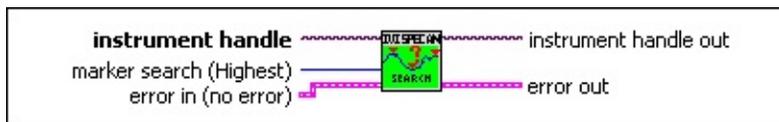
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Marker Search [MKR]

IviSpecAnMarker Capability Group

This VI specifies the type of marker search and performs the search.

This VI returns the Marker Not Enabled error (0x0xBFFA2001) if the IviSpecAn>>Markers>>Marker Enabled property is set to FALSE.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **marker search** Specifies the type of marker search to perform.

Valid Values:

IVISPECAN_VAL_MARKER_SEARCH_HIGHEST
- Sets marker search for the highest amplitude.

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK
- Sets marker search for the next highest peak.

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK_LEFT
- Sets marker search for the next peak left of the peak amplitude.

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEAK_RIGHT
- Sets marker search for the next peak right of the peak amplitude.

IVISPECAN_VAL_MARKER_SEARCH_MINIMUM
- Sets marker search for the minimum amplitude.

Default Value: IVISPECAN_VAL_MARKER_SEARCH_MINIMUM

Err [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from

other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



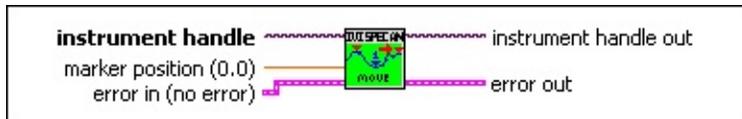
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Move Marker [MKR]

IviSpecAnMarker Capability Group

This VI moves the active marker to the specified horizontal position.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.
- DBL** **marker position** Specifies the frequency or time position of the active marker. The driver uses this value to set the IviSpecAn>>Markers>>Marker Position property.

Units: Hertz or Seconds

Default Value: 0.0

- ERR** **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

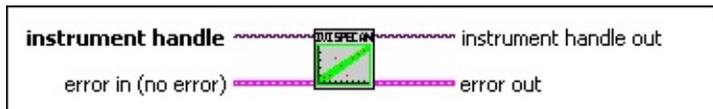
- I/O** **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.
- ERR** **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Peak Preselector [PS]

IviSpecAnPreselector Capability Group

This VI adjusts the preselector to obtain the maximum readings for the current start and stop frequency. This VI may affect the marker configuration.



 **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Query Marker Type [MKT]

IviSpecAnMarkerType Capability Group

This VI queries the read-only IviSpecAn>>Markers>>Marker Type>>Marker Type property.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **marker type** Returns the marker type. This value corresponds to the IviSpecAn>>Markers>>Marker Type>>Marker Type property.

Valid Values:

IVISPECAN_VAL_MARKER_TYPE_NORMAL (1) - Regular marker used to make absolute measurements.

IVISPECAN_VAL_MARKER_TYPE_DELTA (2) - Marker used in conjunction with the reference marker to make relative measurements.

Err **error out** The error out cluster passes error or warning information

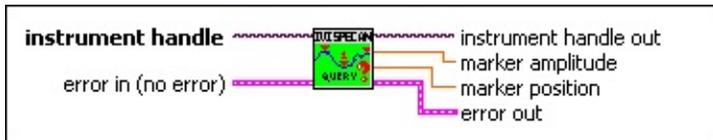
out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Query Marker [MKR]

IviSpecAnMarker Capability Group

This VI returns the horizontal position and the marker amplitude level of the active marker.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

Err **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **marker position** Returns the horizontal position of the active marker from the IviSpecAn>>Markers>>Marker Position property.

DBL **marker amplitude** Returns the amplitude value of the active marker from the IviSpecAn>>Markers>>Marker Amplitude property.

Units: The units are specified by the property, except when the IviSpecAn>>Markers>>Marker Type>>Marker Type property is set to Delta. When IviSpecAn>>Markers>>Marker Type>>Marker Type is set to Delta the units are dB.

Err **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

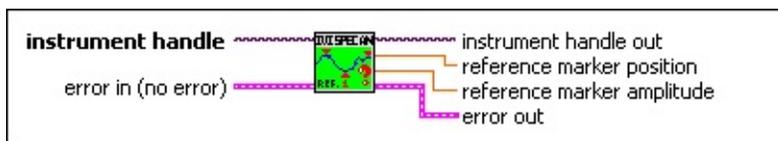
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Query Reference Marker [DMK]

IviSpecAnDeltaMarker Capability Group

This VI returns the amplitude and position of the reference marker.

If the IviSpecAn>>Markers>>Marker Type>>Marker Type property is not Delta, this VI returns the Not Delta Marker error (0xBFFA2002).



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

DBL **reference marker amplitude** Returns the reference marker when the active marker is a delta marker. This returns the value of the IviSpecAn>>Markers>>Delta Marker>>Reference Marker Amplitude property.

Default Value: None

DBL **reference marker position** Returns the reference marker position when the active marker is a delta marker. This returns the value of the IviSpecAn>>Markers>>Delta Marker>>Reference Marker Position property.

Default Value: None



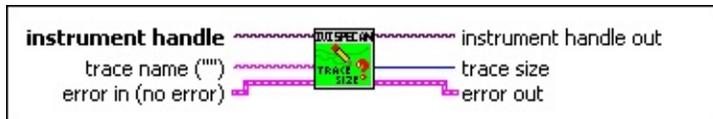
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Query Trace Size

IviSpecAnBase Capability Group

This VI queries the read-only property.



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **trace name** Pass the name of the trace which you request the trace size.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

EH **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32

trace size Returns the number of points in the trace array. This value corresponds to the property.



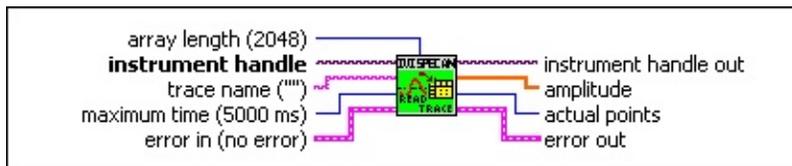
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Read Y Trace

IviSpecAnBase Capability Group

This VI initiates a signal acquisition based on the present instrument configuration. It then waits for the acquisition to complete, and returns the trace as an array of amplitude values. The amplitude array returns data that represent the amplitude of the signals of the sweep from the start frequency to the stop frequency (in frequency domain, in time domain the amplitude array is ordered from beginning of sweep to end). This VI resets the sweep count.



Notes

(1) If the spectrum analyzer did not complete the acquisition within the time specified in the Maximum Time parameter, this VI returns the Max Time Exceeded (0xBFFA2003) error.

(2) If the data was captured in an uncalibrated spectrum analyzer mode, this VI returns a Measure Uncalibrated (0x3FFA2001) warning.

I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **trace name** Pass the name of the trace for the desired amplitude array.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can

differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

 **maximum time (ms)** Pass the maximum length of time in which to allow the read trace operation to complete.

If the operation does not complete within this time interval, the VI returns the `IVISPECAN_ERROR_MAX_TIME_EXCEEDED` (0xBFFA2003) error code. When this occurs, call to cancel the read trace operation and return the spectrum analyzer to the Idle state.

Units: milliseconds.

Defined Values:

`IVISPECAN_VAL_MAX_TIME_INFINITE` (-1)

`IVISPECAN_VAL_MAX_TIME_IMMEDIATE` (0)

Default Value: 5000 (ms)



Notes

(1) The Maximum Time parameter applies only to this VI. It has no effect on other timeout parameters or properties.

 **array length** Pass the number of Amplitude array points requested.

Default Value: None

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **actual points** Indicates the number of points the VI places in the Amplitude array.

 **amplitude** Returns the amplitude waveform that the spectrum analyzer acquires.

Units: The units are determined by setting the Amplitude Units parameter of the or by setting property.

Notes

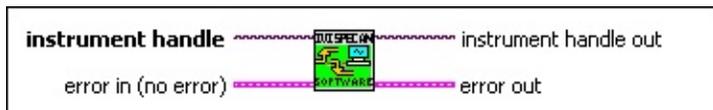
(1) The returns the number of points the spectrum analyzer acquires. Pass a ViReal64 array with at least this many elements.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Send Software Trigger [SWT]

This VI sends a command to trigger the spectrum analyzer. Call this VI if you pass `IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE` for the IviSpecAn>>Trigger>>Trigger Source property or the Trigger Source parameter of the .



Notes

(1) If the IviSpecAn>>Trigger>>Trigger Source is not set to the `IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE` value, this VI returns the Trigger Not Software error (0xBFFA1001).

(2) This VI does not check the instrument status. Typically, you call this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. You use the low-level VIs to optimize one or more aspects of interaction with the instrument. If you want to check the instrument status, call the at the conclusion of the sequence.

 **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



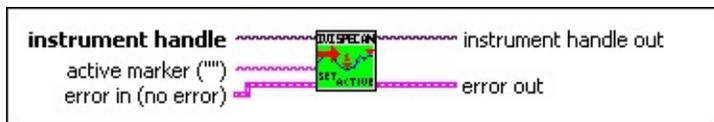
[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Set Active Marker [MKR]

IviSpecAnMarker Capability Group

This VI selects one of the available markers, and makes it the active marker. The active marker must be enabled using the before it can be used for most marker operations.



- I/O** **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.
- abc** **active marker** Pass the name of the marker to be selected. The driver uses this value to set the IviSpecAn>>Markers>>Active Marker property.

This control accepts either a virtual repeated capability name or an instrument-specific marker name.

Virtual repeated capability names are aliases for instrument-specific marker strings. The instrument-specific marker strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

- E.H.** **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more

information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

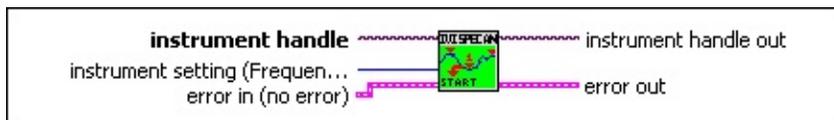
IviSpecAn Set Instrument From Marker [MKR]

IviSpecAnMarker Capability Group

This VI uses the IviSpecAn>>Markers>>Marker Position or IviSpecAn>>Markers>>Marker Amplitude property to configure the spectrum analyzer according to value of the InstrumentSetting parameter. For example, setting the Instrument Setting parameter to Frequency Center sets the center frequency to the value of the IviSpecAn>>Markers>>Marker Position property.

This VI may set the , , or properties.

If the IviSpecAn>>Markers>>Marker Enabled property is set to FALSE, this VI returns the Marker Not Enabled error (0xBFFA2001). If the IviSpecAn>>Markers>>Marker Type>>Marker Type property is not Delta and the InstrumentSetting parameter is Frequency Span, the VI returns the Not Delta Marker error (0xBFFA2002).



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

I32 **instrument setting** Specifies the instrument setting to be set from the marker position.

Defined Values:

Frequency Center -

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_CENTE

Frequency Span -

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_SPAN

Frequency Start -

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_START

Frequency Stop -

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_STOP

Reference Level -

IVISPECAN_VAL_INSTRUMENT_SETTING_REFERENCE_LEVEL

Default Value:

Frequency Center -

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQUENCY_CENTE

 [error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

 [error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

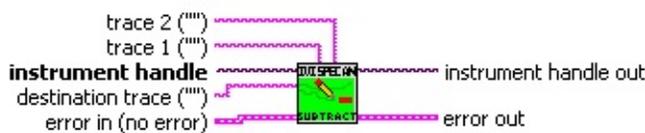
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Subtract Traces [MT]

IviSpecAnMultitrace Capability Group

This VI subtracts the array elements of Trace 2 from Trace 1 and stores the result in the Destination Trace.

Destination Trace = Trace 1 - Trace 2



I/O **instrument handle** The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.

abc **destination trace** Pass the trace name into which the subtracted traces are stored.

Destination Trace = Trace1 - Trace2.

Any data in the destination trace is overwritten.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""

abc **trace 1** Pass the trace name of one of the two traces to be subtracted.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""



trace 2 Pass the trace name of the second trace to be subtracted.

This control accepts either a virtual repeated capability name or an instrument-specific trace name.

Virtual repeated capability names are aliases for instrument-specific trace strings. The instrument-specific trace strings can differ from one instrument to another. Virtual repeated capability names allow you to use and swap instruments without having to change the trace names in your source code. You specify virtual repeated capability names in MAX.

Default Value: ""



[error in \(no error\)](#) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSpecAn Initialize](#) or [IviSpecAn Initialize With Options](#) VIs.



[error out](#) The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSpecAn Properties

Expand this book to view an alphabetized list of IviSpecAn properties.

Active Trace

Active Trace Property

Data type	Access	Applies to	Coercion	High Level VIs
	WO	n/a	none	none

Description

Specifies the trace name used to access all subsequent properties in this instance of the property node. If the property you are attempting to use is trace-based, you must first select the Active Trace property and pass the name of a specific trace. You can pass one of the trace strings that the specific instrument driver defines, or a virtual name the end-user defines in the IVI configuration file in MAX.

Property Node Path

IviSpecAn»Active Trace

IviSpecAn Active Marker [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
-----------	--------	------------	----------	----------------

	R/W	N/A	None	IviSpecAn Set Active Marker [MKR]
-----------------------------------------------------------------------------------	-----	-----	------	---------------------------------------------------

Description

Specifies the marker which is currently active. The values for this property correspond to the Marker repeated capability.

Property Node Path

IviSpecAn»Markers>>Active Marker [MKR]

IviSpecAn Amplitude Units

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Level

Description

Specifies the amplitude units for input, output, and display amplitude.

Property Node Path

IviSpecAn»Basic Operation>>Amplitude Units

Defined Values

IVISPECAN_VAL_AMPLITUDE_UNITS_DBM

IVISPECAN_VAL_AMPLITUDE_UNITS_DBMV

IVISPECAN_VAL_AMPLITUDE_UNITS_DBUV

IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT

IVISPECAN_VAL_AMPLITUDE_UNITS_WATT

IviSpecAn Attenuation

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	Up	IviSpecAn Configure Level

Description

Specifies the input attenuation (in positive dB). This means that if 10dB is selected, the result is a reduction in the signal level of 10 dB.

Property Node Path

IviSpecAn»Basic Operation>>Attenuation

IviSpecAn Attenuation Auto

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Level

Description

Enables or disables auto attenuation.

Property Node Path

IviSpecAn»Basic Operation>>Attenuation Auto

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Bias [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure External Mixer Bias [EM]

Description

Specifies the external mixer bias current in Amps.

Property Node Path

IviSpecAn»External Mixing>>Bias [EM]

IviSpecAn Bias Enabled [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure External Mixer Bias Enabled [EM]

Description

Enables the external mixer's bias.

Property Node Path

IviSpecAn»External Mixing>>Bias Enabled [EM]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Bias Limit [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure External Mixer Bias [EM]

Description

Specifies the external mixer bias current limit in Amps.

Property Node Path

IviSpecAn»External Mixing>>Bias Limit [EM]

IviSpecAn Detector Type

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Acquisition

Description

Specifies the detection method used to capture and process the signal. This governs the data acquisition for a particular sweep, but does not have any control over how multiple sweeps are processed.

Property Node Path

IviSpecAn»Basic Operation>>Detector Type

Defined Values

IVISPECAN_VAL_DETECTOR_TYPE_AUTO_PEAK

IVISPECAN_VAL_DETECTOR_TYPE_AVERAGE

IVISPECAN_VAL_DETECTOR_TYPE_MAX_PEAK

IVISPECAN_VAL_DETECTOR_TYPE_MIN_PEAK

IVISPECAN_VAL_DETECTOR_TYPE_SAMPLE

IVISPECAN_VAL_DETECTOR_TYPE_RMS

IviSpecAn Detector Type Auto

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 TFI	R/W	N/A	None	IviSpecAn Configure Acquisition

Description

Enables the auto detector.

Note:

When the [IviSpecAn>>Basic Operation>>Detector Type](#) Auto property is set to True, the relationship between the [IviSpecAn>>Basic Operation>>Trace Type](#) property and the [IviSpecAn>>Basic Operation>>Detector Type](#) property is not defined.

Property Node Path

IviSpecAn»Basic Operation>>Detector Type Auto

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn External Mixer Average Conversion Loss [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure External Mixer [EM]

Description

Specifies the average conversion loss.

Property Node Path

IviSpecAn»External Mixing>>External Mixer Average Conversion Loss
[EM]

IviSpecAn External Mixer Conversion Loss Table Enabled [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Conversion Loss Table Enabled [EM]

Description

Enables or disables the conversion loss table.

Property Node Path

IviSpecAn»External Mixing>>External Mixer Conversion Loss Table
Enabled [EM]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn External Mixer Enabled [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure External Mixer Enabled [EM]

Description

Enables the external mixer.

Property Node Path

IviSpecAn»External Mixing>>External Mixer Enabled [EM]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn External Mixer Harmonic [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure External Mixer [EM]

Description

Specifies the order n of the harmonic used for conversion.

Property Node Path

IviSpecAn»External Mixing>>External Mixer Harmonic [EM]

IviSpecAn External Mixer Number of Ports [EM]

IviSpecAnExternalMixer Property

Data type	Access	Applies to	Coercion	High Level VIs
 I32	R/W	N/A	None	None

Description

Selects the number of ports used in external mixing.

Property Node Path

IviSpecAn»External Mixing>>External Mixer Number of Ports [EM]

IviSpecAn External Trigger Level [EXT]

IviSpecAnExternalTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure External Trigger [EXT]

Description

Specifies the level, in Volts, of the external trigger signal to trigger an acquisition.

Property Node Path

IviSpecAn»Trigger>>External Trigger>>External Trigger Level [EXT]

IviSpecAn External Trigger Slope [EXT]

IviSpecAnExternalTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure External Trigger [EXT]

Description

Specifies the slope of the external trigger signal to trigger an acquisition.

Property Node Path

IviSpecAn»Trigger>>External Trigger>>External Trigger Slope [EXT]

Defined Values

IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_POSITIVE

IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_NEGATIVE

IviSpecAn Frequency Offset

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Frequency Offset

Description

Specifies an offset value, in Hertz, that is added to the frequency readout. The offset is used to compensate for external frequency conversion.

Setting this property changes the [IviSpecAn>>Basic Operation>>Frequency Start](#) and [IviSpecAn>>Basic Operation>>Frequency Stop](#) properties. The following equations define the relationship of these settings:

Frequency Start = Actual Start Frequency + Frequency Offset

Frequency Stop = Actual Stop Frequency + Frequency Offset

Marker Position = Actual Marker Frequency + Frequency Offset

Property Node Path

IviSpecAn»Basic Operation>>Frequency Offset

IviSpecAn Frequency Start

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Frequency Start Stop IviSpecAn Configure Frequency Center Span

Description

Specifies the left edge of the frequency domain in Hertz. This is used in conjunction with the [IviSpecAn>>Basic Operation>>Frequency Stop](#) property to define the frequency domain. If the [IviSpecAn>>Basic Operation>>Frequency Start](#) property value is equal to the [IviSpecAn>>Basic Operation>>Frequency Stop](#) property value then the spectrum analyzer's horizontal properties are in time-domain.

Property Node Path

IviSpecAn»Basic Operation>>Frequency Start

IviSpecAn Frequency Stop

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	<u>IviSpecAn Configure Frequency Start Stop</u> <u>IviSpecAn Configure Frequency Center Span</u>

Description

Specifies the right edge of the frequency domain in Hertz. This is used in conjunction with the [IviSpecAn>>Basic Operation>>Frequency Stop](#) property to define the frequency domain. If the [IviSpecAn>>Basic Operation>>Frequency Start](#) property value is equal to the [IviSpecAn>>Basic Operation>>Frequency Stop](#) property value then the spectrum analyzer's horizontal properties are in time-domain.

Property Node Path

IviSpecAn»Basic Operation>>Frequency Stop

IviSpecAn Input Impedance

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Level

Description

Specifies the value of input impedance, in ohms, expected at the active input port. This is typically 50 ohms or 75 ohms.

Property Node Path

IviSpecAn»Basic Operation>>Input Impedance

IviSpecAn Marker Amplitude [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	IviSpecAn Query Marker [MKR]

Description

Returns the amplitude of the active marker. The units are specified by the [IviSpecAn>>Basic Operation>>Amplitude Units](#) property, except when the [IviSpecAn>>Markers>>Marker Type>>Marker Type \[MKR\]](#) property is set to Delta. When the [IviSpecAn>>Markers>>Marker Type>>Marker Type \[MKR\]](#) property is set to Delta the units are dB. If the [IviSpecAn>>Markers>>Marker Enabled \[MKR\]](#) property is set to False, any attempt to read this property returns the Marker Not Enabled error(0xBFFA2001).

Property Node Path

IviSpecAn»Markers>>Marker Amplitude [MKR]

IviSpecAn Marker Enabled [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Marker Enabled [MKR]

Description

Enables the active marker.

Property Node Path

IviSpecAn»Markers>>Marker Enabled [MKR]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Marker Frequency Counter Enabled [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Marker Frequency Counter [MKR]

Description

Enables the marker frequency counter for better marker measurement accuracy. This property returns the Marker Not Enabled error (0xBFFA2001) if the [IviSpecAn>>Markers>>Marker Enabled \[MKR\]](#) property is set to False.

Property Node Path

IviSpecAn»Markers>>Marker Frequency Counter Enabled [MKR]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Marker Frequency Counter Resolution [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	Down	IviSpecAn Configure Marker Frequency Counter [MKR]

Description

Specifies the resolution of the frequency counter in Hertz. The measurement gate time is the reciprocal of the specified resolution.

Property Node Path

IviSpecAn»Markers>>Marker Frequency Counter Resolution [MKR]

IviSpecAn Marker Position [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Move Marker [MKR] IviSpecAn Query Marker [MKR]

Description

Specifies the frequency or time position of the active marker (depending on the mode in which the analyzer is operating, frequency or time-domain). This property returns the Marker Not Enabled error (0xBFFA2001) if the active marker is not enabled.

Property Node Path

IviSpecAn»Markers>>Marker Position [MKR]

IviSpecAn Marker Threshold [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Marker Search [MKR]

Description

Specifies the lower limit of the search domain vertical range for the [IviSpecAn Marker Search \[MKR\].vi](#).

Property Node Path

IviSpecAn»Markers>>Marker Threshold [MKR]

IviSpecAn Marker Trace [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Marker Enabled [MKR]

Description

Specifies the trace for the active marker.

Property Node Path

IviSpecAn»Markers>>Marker Trace [MKR]

IviSpecAn Marker Type [MKT]

IviSpecAnMarkerType Property

Data type	Access	Applies to	Coercion	High Level VIs
I32i	RO	N/A	None	IviSpecAn Query Marker Type [MKT]

Description

Returns the marker type of the active marker.

Property Node Path

IviSpecAn»Markers>>Marker Type>>Marker Type [MKT]

Defined Values

IVISPECAN_VAL_MARKER_TYPE_NORMAL

IVISPECAN_VAL_MARKER_TYPE_DELTA

IviSpecAn Number of Divisions [DSP]

IviSpecAnDisplay Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Specifies the number of divisions.

Property Node Path

IviSpecAn»Display Control>>Number of Divisions [DSP]

IviSpecAn Number of Markers [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Returns the number of markers available for the instrument.

Property Node Path

IviSpecAn»Markers>>Number of Markers [MKR]

IviSpecAn Number of Sweeps

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Acquisition

Description

This property defines the number of sweeps. This property value has no effect if the [IviSpecAn>>Basic Operation>>Trace Type](#) property is set to the value Clear Write.

Property Node Path

IviSpecAn»Basic Operation>>Number of Sweeps

IviSpecAn Peak Excursion [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Marker Search [MKR]

Description

Specifies the minimum amplitude variation of the signal in dB that the [IviSpecAn Marker Search \[MKR\].vi](#) identifies as a peak.

Property Node Path

IviSpecAn»Markers>>Peak Excursion [MKR]

IviSpecAn Reference Level

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Level

Description

The calibrated vertical position of the captured data used as a reference for amplitude measurements. This is typically set to a value slightly higher than the highest expected signal level. The units are determined by the [IviSpecAn>>Basic Operation>>Amplitude Units](#) property.

Property Node Path

IviSpecAn»Basic Operation>>Reference Level

IviSpecAn Reference Level Offset

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Level

Description

Specifies an offset for the [IviSpecAn>>Basic Operation>>Reference Level](#) property in dB. This property adjusts the reference level for external signal gain or loss. A positive value corresponds to a gain while a negative number corresponds to a loss.

Property Node Path

IviSpecAn»Basic Operation>>Reference Level Offset

IviSpecAn Reference Marker Amplitude [DMK]

IviSpecAnDeltaMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	IviSpecAn Query Reference Marker [DMK]

Description

Returns the reference marker amplitude when the active marker is a delta marker. The units are specified by the [IviSpecAn>>Basic Operation>>Amplitude Units](#) property. If the Marker Type property is not Delta, this property returns the Not Delta Marker error (0xBFFA2002).

Property Node Path

IviSpecAn»Markers>>Delta Marker>>Reference Marker Amplitude [DMK]

IviSpecAn Reference Marker Position [DMK]

IviSpecAnDeltaMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	RO	N/A	None	IviSpecAn Query Reference Marker [DMK]

Description

Returns the position of the reference marker, when the active marker is a delta marker. The units are Hertz for frequency domain measurements, and seconds for time domain measurements. If the Marker Type property is not Delta, this property returns the Not Delta Marker error (0xBFFA2002).

Property Node Path

IviSpecAn»Markers>>Delta Marker>>Reference Marker Position [DMK]

IviSpecAn Resolution Bandwidth

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**

 R/W N/A Up [IviSpecAn Configure Sweep Coupling](#)

Description

This specifies the width of the IF filter in Hertz.

Property Node Path

IviSpecAn»Basic Operation>>Resolution Bandwidth

IviSpecAn Resolution Bandwidth Auto

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**

 R/W N/A None [IviSpecAn Configure Sweep Coupling](#)

Description

Enables resolution bandwidth auto coupling.

Property Node Path

IviSpecAn»Basic Operation>>Resolution Bandwidth Auto

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Signal Track Enabled [MKR]

IviSpecAnMarker Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Signal Track Enabled [MKR]

Description

Enables or disables signal tracking. When signal tracking is turned on, the spectrum analyzer centers the signal after each sweep. This process invalidates the the [IviSpecAn>>Basic Operation>>Frequency Start](#) and [IviSpecAn>>Basic Operation>>Frequency Stop](#) properties. If the active marker is not enabled, operations on this property return the Marker Not Enabled (0xBFFA2001) error.

Note: Signal tracking can only be enabled on one marker at any given time.

Property Node Path

IviSpecAn»Markers>>Signal Track Enabled [MKR]

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Sweep Mode Continuous

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
 Boolean	R/W	N/A	None	IviSpecAn Configure Acquisition

Description

Enables continuous sweep mode.

Property Node Path

IviSpecAn»Basic Operation>>Sweep Mode Continuous

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Sweep Time

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**



R/W

N/A

Up

[IviSpecAn Configure Sweep Coupling](#)

Description

Specifies the length of time (in seconds) to sweep from the left edge to the right edge of the current domain.

Property Node Path

IviSpecAn»Basic Operation>>Sweep Time

IviSpecAn Sweep Time Auto

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**

 R/W N/A None [IviSpecAn Configure Sweep Coupling](#)

Description

Enables sweep time auto.

Property Node Path

IviSpecAn»Basic Operation>>Sweep Time Auto

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Trace Count

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	N/A	None	None

Description

Returns the number of traces.

Note:

For an instrument with only one Trace the driver may return an empty string.

Property Node Path

IviSpecAn»Basic Operation>>Trace Count

IviSpecAn Trace Size

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	RO	Trace	None	IviSpecAn Query Trace Size

Description

This trace-based property returns the number of points in the trace array.

Property Node Path

IviSpecAn»Basic Operation>>Trace Size

IviSpecAn Trace Type

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	Trace	None	IviSpecAn Configure Trace Type

Description

This trace-based property specifies the representation of the acquired data.

Property Node Path

IviSpecAn»Basic Operation>>Trace Type

Defined Values

IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE

IVISPECAN_VAL_TRACE_TYPE_MAX_HOLD

IVISPECAN_VAL_TRACE_TYPE_MIN_HOLD

IVISPECAN_VAL_TRACE_TYPE_VIDEO_AVERAGE

IVISPECAN_VAL_TRACE_TYPE_VIEW

IVISPECAN_VAL_TRACE_TYPE_STORE

IviSpecAn Trigger Source [TRG]

IviSpecAnTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Trigger Source [TRG]

Description

Specifies the source of the trigger signal that causes the analyzer to leave the Wait-For-Trigger state.

Property Node Path

IviSpecAn»Trigger>>Trigger Source [TRG]

Defined Values

IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNAL

IVISPECAN_VAL_TRIGGER_SOURCE_IMMEDIATE

IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE

IVISPECAN_VAL_TRIGGER_SOURCE_AC_LINE

IVISPECAN_VAL_TRIGGER_SOURCE_VIDEO

IviSpecAn Units Per Division [DSP]

IviSpecAnDisplay Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	None

Description

Specifies the number of vertical units in one screen division. This is typically used in conjunction with the [IviSpecAn>>Basic Operation>>Reference Level](#) property to set the vertical range of the spectrum analyzer.

Property Node Path

IviSpecAn»Display Control>>Units Per Division [DSP]

IviSpecAn Vertical Scale

IviSpecAnBase Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Acquisition

Description

Specifies the vertical scale of the measurement hardware (use of log amplifiers versus linear amplifiers).

Property Node Path

IviSpecAn»Basic Operation>>Vertical Scale

Defined Values

IVISPECAN_VAL_VERTICAL_SCALE_LINEAR

IVISPECAN_VAL_VERTICAL_SCALE_LOGARITHMIC

IviSpecAn Video Bandwidth

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**

 R/W N/A Up [IviSpecAn Configure Sweep Coupling](#)

Description

Specifies the video bandwidth of the post-detection filter in Hertz.

Property Node Path

IviSpecAn»Basic Operation>>Video Bandwidth

IviSpecAn Video Bandwidth Auto

IviSpecAnBase Property

Data type **Access** **Applies to** **Coercion** **High Level VIs**

 R/W N/A None [IviSpecAn Configure Sweep Coupling](#)

Description

Enables video bandwidth auto coupling.

Property Node Path

IviSpecAn»Basic Operation>>Video Bandwidth Auto

Defined Values

VI_TRUE

VI_FALSE

IviSpecAn Video Trigger Level [VT]

IviSpecAnVideoTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
 DBL	R/W	N/A	None	IviSpecAn Configure Video Trigger [VT]

Description

Specifies the level of the video signal to trigger an acquisition. The units are specified by the [IviSpecAn>>Basic Operation>>Amplitude Units](#) property.

Property Node Path

IviSpecAn»Trigger>>Video Trigger>>Video Trigger Level [VT]

IviSpecAn Video Trigger Slope [VT]

IviSpecAnVideoTrigger Property

Data type	Access	Applies to	Coercion	High Level VIs
	R/W	N/A	None	IviSpecAn Configure Video Trigger [VT]

Description

Specifies the slope of the video signal to trigger an acquisition.

Property Node Path

IviSpecAn»Trigger>>Video Trigger>>Video Trigger Slope [VT]

Defined Values

IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POSITIVE

IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_NEGATIVE

IviSpecAn Defined Values

Expand this book to view an alphabetized list of IviSpecAn defined values.

IVISPECAN_VAL_AMPLITUDE_UNITS_DBM

Description

Sets the spectrum Analyzer to measure in decibels relative to 1 milliwatt.

Defined Value

1

IVISPECAN_VAL_AMPLITUDE_UNITS_DBMV

Description

Sets the spectrum analyzer to measure in decibels relative to 1 millivolt.

Defined Value

2

IVISPECAN_VAL_AMPLITUDE_UNITS_DBUV

Description

Sets the spectrum analyzer to measure in decibels relative to 1 microvolt.

Defined Value

3

IVISPECAN_VAL_AMPLITUDE_UNITS_VOLT

Description

Sets the spectrum analyzer to measure in volts.

Defined Value

4

IVISPECAN_VAL_AMPLITUDE_UNITS_WATT

Description

Sets the spectrum analyzer to measure in watts.

Defined Value

5

IVISPECAN_VAL_DETECTOR_TYPE_AUTO_PEA

Description

Allows the detector to capture better readings by using both positive and negative peak values when noise is present.

Defined Value

1

IVISPECAN_VAL_DETECTOR_TYPE_AVERAGE

Description

Average value of samples taken within the bin for a dedicated point on the display.

Defined Value

2

IVISPECAN_VAL_DETECTOR_TYPE_MAX_PEAK

Description

Obtains the maximum video signal between the last display point and the present display point.

Defined Value

3

IVISPECAN_VAL_DETECTOR_TYPE_MIN_PEAK

Description

Obtains the minimum video signal between the last display point and the present display point.

Defined Value

4

IVISPECAN_VAL_DETECTOR_TYPE_RMS

Description

RMS value of samples taken within the bin for a dedicated point on the display.

Defined Value

6

IVISPECAN_VAL_DETECTOR_TYPE_SAMPLE

Description

Pick one point within a bin.

Defined Value

5

IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_

Description

Sets negative slope.

Defined Value

2

IVISPECAN_VAL_EXTERNAL_TRIGGER_SLOPE_

Description

Sets positive slope.

Defined Value

1

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQ

Description

Sets the center frequency with the Marker Position attribute.

Defined Value

1

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQ

Description

Sets the frequency span with the Marker Position attribute. This operation is only valid if the Marker Type attribute is set to Delta.

Defined Value

2

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQ

Description

Sets the Frequency Start attribute with the Marker Position attribute.

Defined Value

3

IVISPECAN_VAL_INSTRUMENT_SETTING_FREQ

Description

Sets the Frequency Stop attribute with the Marker Position attribute.

Defined Value

4

IVISPECAN_VAL_INSTRUMENT_SETTING_REFER

Description

Sets the Reference Level attribute with the Marker Amplitude attribute.

Defined Value

5

IVISPECAN_VAL_MARKER_SEARCH_HIGHEST

Description

Sets marker search for the highest amplitude.

Defined Value

1

IVISPECAN_VAL_MARKER_SEARCH_MINIMUM

Description

Sets marker search for the minimum amplitude.

Defined Value

2

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEA

Description

Sets marker search for the next highest peak.

Defined Value

3

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEA

Description

Sets marker search for the next peak left of the peak amplitude.

Defined Value

4

IVISPECAN_VAL_MARKER_SEARCH_NEXT_PEA

Description

Sets marker search for the next peak right of the peak amplitude.

Defined Value

5

IVISPECAN_VAL_MARKER_TYPE_DELTA

Description

Marker used in conjunction with the reference marker to make relative measurements.

Defined Value

2

IVISPECAN_VAL_MARKER_TYPE_NORMAL

Description

Regular marker used to make absolute measurements.

Defined Value

1

IVISPECAN_VAL_TRACE_TYPE_CLEAR_WRITE

Description

Sets the spectrum analyzer to clear previous sweep data off the display before performing a sweep. Subsequent sweeps may or may not clear the display first, but the data array at the end of the sweep is entirely new.

Defined Value

1

IVISPECAN_VAL_TRACE_TYPE_MAX_HOLD

Description

Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, which ever is higher.

Defined Value

2

IVISPECAN_VAL_TRACE_TYPE_MIN_HOLD

Description

Sets the spectrum analyzer to keep the data from either the previous data or the new sweep data, which ever is lower.

Defined Value

3

IVISPECAN_VAL_TRACE_TYPE_STORE

Description

Disables acquisition and disables the display of the existing trace data.

Defined Value

6

IVISPECAN_VAL_TRACE_TYPE_VIDEO_AVERAG

Description

Sets the spectrum analyzer to maintain a running average of the swept data.

Defined Value

4

IVISPECAN_VAL_TRACE_TYPE_VIEW

Description

Disables acquisition into this trace but displays the existing trace data.

Defined Value

5

IVISPECAN_VAL_TRIGGER_SOURCE_AC_LINE

Description

The spectrum analyzer waits until it receives a trigger on the AC line.

Defined Value

4

IVISPECAN_VAL_TRIGGER_SOURCE_EXTERNA

Description

The spectrum analyzer waits until it receives a trigger on the external trigger connector.

Defined Value

1

IVISPECAN_VAL_TRIGGER_SOURCE_IMMEDIAT

Description

The spectrum analyzer does not wait for a trigger of any kind.

Defined Value

2

IVISPECAN_VAL_TRIGGER_SOURCE_SOFTWARE

Description

The spectrum analyzer waits until the Send Software Trigger function executes.

Defined Value

3

IVISPECAN_VAL_TRIGGER_SOURCE_VIDEO

Description

The spectrum analyzer waits until it receives a video level.

Defined Value

5

IVISPECAN_VAL_VERTICAL_SCALE_LINEAR

Description

Sets the vertical scale in linear units.

Defined Value

1

IVISPECAN_VAL_VERTICAL_SCALE_LOGARITH

Description

Sets the vertical scale in logarithmic units.

Defined Value

2

IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_NEG

Description

Sets negative slope.

Defined Value

2

IVISPECAN_VAL_VIDEO_TRIGGER_SLOPE_POS

Description

Sets positive slope.

Defined Value

1



IviSwtch Class Driver Reference for LabVIEW

Concepts

[IviSwtch Terminology](#)

[IviSwtch Class Driver Overview](#)

[IviSwtch Extension Capability Groups](#)

[IviSwtch Behavior Models](#)

IviSwtch VI and Property Reference

[IviSwtch VI Tree](#)

[IviSwtch Properties by Group](#)

[IviSwtch Property Value Definitions](#)

[IviSwtch Error and Completion Codes](#)

Other Reference

[Interchangeability Checking](#)

[Simulation](#)

IviSwtch Terminology

Channel	An input/output (I/O) connection on the switch module that you can access. What constitutes a channel is defined by the vendor, but the channel must be a point that you can connect to one or more other channels of the switch module through a path. In addition, it is the connection point to the switch module. Notice that a channel does not indicate the number of wires. A channel may consist of 1, 2, 3 or 4 wires, for example.
Channel Pair	Two channel names separated by the "->" symbol.
Common	The name of the output channel in a multiplexer switch module.
Configuration Channel	A channel that is not directly accessible to you through the IviSwtch class driver, or that you mark as a configuration channel to reserve for the path creation. The driver uses a configuration channel to create paths between the channels, connect or disconnect to an analog bus, and so on. This gives the driver more flexibility in creating paths at the expense of losing channels. Mark a column in a matrix as a configuration channel when you want to allow the matrix to connect a row to a row.
Matrix Switch Module	A switch module that is configured to have multiple inputs and outputs that form a standard matrix organization such that any row can be connected to any column. Notice that some, but not all matrices support row-to-row and column-to-column connections. See Configuration Channel.
Multiplexer Switch Module	A switch module that is configured to have multiple input channels but only a single output channels. Other names for the multiplexer switch module are a tree, and a 1 x n matrix.
Path	The connection (electrical, optical, etc.) between the two channels. You create a path with operations defined in the IviSwtch class. The end-point channels define such a connection. Notice that it is up to the switch module to

	know what paths are valid, invalid or in use.
Scanner Switch Module	An IviSwTch switch module with the capability to scan channels.
Source Channel	A channel directly accessible to you through the IviSwTch class driver. Typically, the driver marks a channel as a source channel to allow for external connection.
Switch Module	The vendor defined device that the instrument driver session can communicate with and control. The channels of such a device define a switch module. Notice that on a physical switch device there may be multiple switch modules. In addition, a switch module may be on multiple switch devices. The concept is to have a single black box with external connections and have the software find the necessary paths. Notice that this does not remove the need of the application programmer to understand the underlying switch structure for issues such as sending the correct signals through the correct switches (for example, RF signals through RF paths only).

IviSwtch Class Driver Overview

This instrument driver provides programming support for the IviSwtch Class. The IviSwtch class driver conceptualizes a switch as an instrument that can establish a connection between two I/O channels. The IviSwtch class driver can support instruments that can perform triggered scanning (trigger-synchronized establishing or breaking of the paths). The driver contains all the VIs that the IVI Foundation requires for the IviSwtch specification. This driver requires NI-VISA and the IVI Compliance Package to be installed.

An IviSwtch is a vendor-defined switch module with a series of I/O capable channels. These channels can then be connected through the internals of the switch module, where not all connections are necessarily valid. Refer to the figure below. The IviSwtch class driver is designed to support the typical switch as well as common extended functionality found in more complex switch instruments. The IviSwtch class driver conceptualizes the switch as an instrument that can establish paths between its I/O channels.

Use this driver to develop programs that are independent of a particular switch. You can use this class driver with any switch that has an IVI instrument specific driver that is compliant with the IviSwtch class. The IviSwtch class driver accesses the specific driver for your instrument using the configuration information you supply with MAX.

The IviSwtch class driver divides instrument capabilities into a base capability group and multiple extension capability groups. The [IviSwtchBase capability group](#) supports switches able to connect and disconnect paths on the instrument, determine the connectivity of two switches, and query the state of the switch module. The VIs and properties of this group are implemented by all switch instrument drivers that are compliant with the IviSwtch class.

IviSwtch Extension Capability Groups

The IviSwtch class driver identifies VIs and properties that belong to an extension group with the following abbreviations. Instrument specific drivers are not required to implement any of the extension capability groups. If you use VIs and properties that belong to extension capability groups, every instrument specific driver that you use must implement those extensions.

Group Name	Description
IviSwtchScanner [SCN]	Defines extensions for instruments capable of scanning channels.
IviSwtchSoftwareTrigger [SWT]	Defines instruments capable of receiving software triggers and implements the IviSwtchScanner extension group.

IviSwtchScanner defines extended capabilities for switches that can wait for the trigger to establish or break paths on the switch module, and assert a trigger upon completed task. The switches that can perform such tasks are the part of the IviSwtchScanner extension group.

IviSwtchSoftwareTrigger defines extended capabilities for switches that can support IviSwtchScanner extension group and receive software triggers.



Switch Module

IviSwtchBaseCapability Group

The IviSwtchBase capability group supports switch instruments capable of connecting and disconnecting paths within the instrument, determining the connectivity of two switches, and querying various internal states of the switch module. The IviSwtchBase capability group defines properties that reflect the physical characteristics of the switch module, as well as properties that allow you to specify the connectivity of switching channels. The IviSwtchBase capability group also defines a set of VIs that allows you to connect and disconnect paths, as well as query the connectivity of paths.

IviSwtchBase VIs

[IviSwtch Connect Channels](#)

[IviSwtch Disconnect Channels](#)

[IviSwtch Get Path](#)

[IviSwtch Set Path](#)

[IviSwtch Can Connect Channels](#)

[IviSwtch Disconnect All Channels](#)

[IviSwtch Switch Is Debounced](#)

[IviSwtch Wait For Debounce](#)

IviSwchBase Properties

[IviSwch Bandwidth](#)

[IviSwch Characteristic Impedance](#)

[IviSwch Is Configuration Channel](#)

[IviSwch Is Debounced](#)

[IviSwch Is Source Channel](#)

[IviSwch Maximum AC Voltage](#)

[IviSwch Maximum Carry AC Current](#)

[IviSwch Maximum Carry AC Power](#)

[IviSwch Maximum Carry DC Current](#)

[IviSwch Maximum Carry DC Power](#)

[IviSwch Maximum DC Voltage](#)

[IviSwch Maximum Switching AC Current](#)

[IviSwch Maximum Switching AC Power](#)

[IviSwch Maximum Switching DC Current](#)

[IviSwch Maximum Switching DC Power](#)

[IviSwch Settling Time](#)

[IviSwch Wire Mode](#)

Related Topics

[VI Tree](#)

[IviSwch Properties](#)

[IVI Inherent VIs](#)

[IVI Inherent Properties](#)

[IviSwchBase behavior model](#)

IviSwtchScanner Extension Group

The IviSwtchScanner Extension Group defines a set of properties and VIs to perform scanning operations.

Refer to the [IviSwtchScanner behavior model](#).

IviSwtchScanner VIs

[IviSwtch Initiate Scan](#)

[IviSwtch Abort Scan](#)

[IviSwtch Switch Is Scanning](#)

[IviSwtch Wait For Scan To Complete](#)

[IviSwtch Configure Scan List](#)

[IviSwtch Configure Scan Trigger](#)

[IviSwtch Set Continuous Scan](#)

IviSwtchScanner Properties

[IviSwtch Continuous Scan](#)

[IviSwtch Is Scanning](#)

[IviSwtch Number of Rows](#)

[IviSwtch Number of Columns](#)

[IviSwtch Scan Advanced Output](#)

[IviSwtch Scan List](#)

[IviSwtch Scan Mode](#)

[IviSwtch Scan Delay](#)

[IviSwtch Trigger Input](#)

Related Topics

[VI Tree](#)

[IviSwtch Properties](#)

IviSwtchSoftwareTrigger Extension Group

The IviSwtchSoftwareTrigger Extension Group defines a set of VIs that perform software trigger operations.

IviSwchSoftwareTrigger VI

[IviSwch Send Software Trigger](#)

Related Topics

[VI Tree](#)

[IviSwch Properties](#)

IviSwtch Behavior Models

[IviSwtchBase](#)

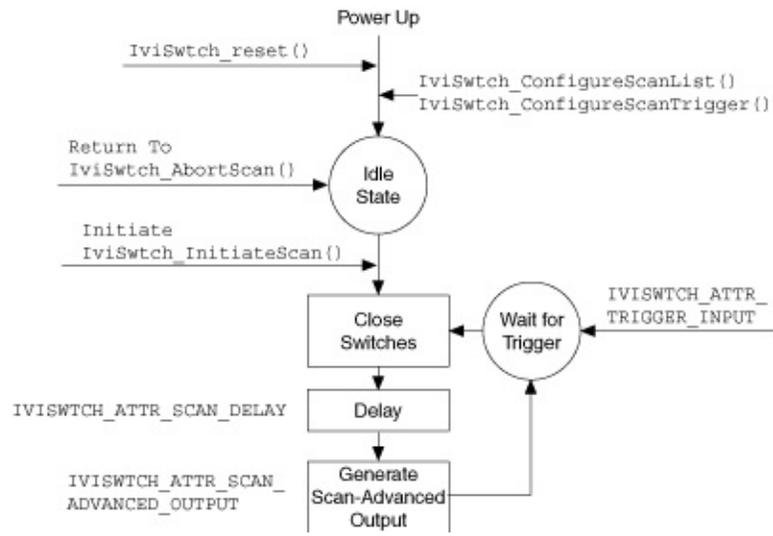
[IviSwtchScanner](#)

[IviSwtchSoftwareTrigger](#)

IviSwtchBase Behavior Model

You can access any of the VIs in this capability group at any time. If you execute the [IviSwtch Wait For Debounce](#) VI, the driver blocks any further operation until the VI completes (that is, all the signals flowing through the switch have settled).

IviSwthScanner Behavior Model



IviSwtchScanner Behavior Model

It is the IVI driver's responsibility to ensure that when the scanning begins a trigger is sent from the switch module if the switch module is configured to assert a trigger on path creation (the [IviSwtch Scan Advanced Output \[SCN\]](#) property). This ensures that if the switch module is using handshake lines with a measurement or source device and also using scanning, the sequence is begun with a trigger from the switch module.

When not in the Idle or Reset state, all properties of the IviSwtch class are read only. Similarly, when not in the Idle or Reset state, properties are read only, and only the [IviSwtch Reset](#) and [IviSwtch Abort Scan](#) properties can be executed.

IviSwtchSoftwareTrigger Behavior Model

The IviSwtchSoftwareTrigger extension group follows the behavior model of the [IviSwtchScanner](#) extension group. If the [IviSwtch Trigger Input \[SCN\]](#) property is set to [IVISWTCH_VAL_SOFTWARE TRIG](#), the switch exits the wait-for-trigger state only after the [IviSwtch Send Software Trigger](#) VI executes.

IviSwtch VI Tree

The VI tree for the IviSwtch class driver, including IVI and VXI*plug&play* required VIs, is shown below.



Note You use the LabVIEW Property Node to get and set properties.

IviSwtch VI Tree

Name or Class	VI Name	Required By
Initialize	<u>IviSwtch Initialize</u>	VPP
Initialize With Options	<u>IviSwtch Initialize With Options</u>	IVI
Configuration VIs		
Configure Scan List [SCN]	<u>IviSwtch Configure Scan List</u>	IviSwtchScanner
Configure Scan Trigger [SCN]	<u>IviSwtch Configure Scan Trigger</u>	IviSwtchScanner
Set Continuous Scan [SCN]	<u>IviSwtch Set Continuous Scan</u>	IviSwtchScanner
Route		
Connect Channels	<u>IviSwtch Connect Channels</u>	IviSwtchBase
Disconnect Channels	<u>IviSwtch Disconnect Channels</u>	IviSwtchBase
Disconnect All Channels	<u>IviSwtch Disconnect All Channels</u>	IviSwtchBase
Switch Is Debounced?	<u>IviSwtch Switch Is Debounced</u>	IviSwtchBase
Wait For Debounce	<u>IviSwtch Wait For Debounce</u>	IviSwtchBase
Can Connect Channels?	<u>IviSwtch Can Connect Channels</u>	IviSwtchBase
Paths		
Set Path	<u>IviSwtch Set Path</u>	IviSwtchBase

Get Path	<u>IviSwtch Get Path</u>	IviSwtchBase
Scan		
Initiate Scan [SCN]	<u>IviSwtch Initiate Scan</u>	IviSwtchScanner
Abort Scan [SCN]	<u>IviSwtch Abort Scan</u>	IviSwtchScanner
Switch Is Scanning [SCN]	<u>IviSwtch Switch Is Scanning</u>	IviSwtchScanner
Wait For Scan To Complete [SCN]	<u>IviSwtch Wait For Scan To Complete</u>	IviSwtchScanner
Send Software Trigger [SWT]	<u>IviSwtch Send Software Trigger</u>	IviSwtchSoftwareTrigger
Utility		
Reset	<u>IviSwtch Reset</u>	VPP
Reset With Defaults	<u>Reset With Defaults</u>	VPP
Self-Test	<u>IviSwtch Self-Test</u>	VPP
Disable	<u>Disable</u>	VPP
Invalidate All Attributes	<u>Invalidate All Attributes</u>	VPP
Revision Query	<u>IviSwtch Revision Query</u>	VPP
Error-Query	<u>IviSwtch Error-Query</u>	VPP
Error Message	<u>IviSwtch Error Message</u>	VPP
Get Channel Name	<u>Get Channel Name</u>	VPP
Interchangeability Info		
Get Next	<u>IviSwtch Get Next</u>	IVI

Interchange
Warning

[Interchange Warning](#)

Reset Interchange
Check

[IviSwtch Reset
Interchange Check](#)

IVI

Clear Interchange
Warnings

[IviSwtch Clear
Interchange Warnings](#)

IVI

Coercion Info

Get Next Coercion
Record

[IviSwtch Get Next
Coercion Record](#)

IVI

Close

[IviSwtch Close](#)

VPP

Related Topics

[IviSwtch Properties](#)

[IVI Inherent VIs](#)

IviSwtch Properties



Note You use the LabVIEW Property Node to get and set properties.

IviSwtch Base Properties

[IviSwtch Bandwidth](#)

[IviSwtch Characteristic Impedance](#)

[IviSwtch Is Configuration Channel](#)

[IviSwtch Is Debounced](#)

[IviSwtch Is Source Channel](#)

[IviSwtch Maximum AC Voltage](#)

[IviSwtch Maximum Carry AC Current](#)

[IviSwtch Maximum Carry AC Power](#)

[IviSwtch Maximum Carry DC Current](#)

[IviSwtch Maximum Carry DC Power](#)

[IviSwtch Maximum DC Voltage](#)

[IviSwtch Maximum Switching AC Current](#)

[IviSwtch Maximum Switching AC Power](#)

[IviSwtch Maximum Switching DC Current](#)

[IviSwtch Maximum Switching DC Power](#)

[IviSwtch Settling Time](#)

[IviSwtch Wire Mode](#)

IviSwthScanner Extension Properties

[IviSwth Continuous Scan](#)

[IviSwth Is Scanning](#)

[IviSwth Number of Rows](#)

[IviSwth Number of Columns](#)

[IviSwth Scan Advanced Output](#)

[IviSwth Scan List](#)

[IviSwth Scan Mode](#)

[IviSwth Scan Delay](#)

[IviSwth Trigger Input](#)

IviSwtch Property Value Definitions

The following table defines value for all IviSwtch class properties constants.



Note In the following table, the literal string IVISWTCH precedes the property name and the literal string IVISWTCH_ precedes the value extension name.

IviSwtch Property Defined Values

Property	Value Name	Defined Value	Description
SCAN MODE	No Action	0	Indicates that no action should be taken on the previous paths.
	Before Break Make	1	Tells the device to break the previous paths before making the new paths.
	Break After Make	2	Tells the driver to make the new paths before breaking the previous paths.
TRIGGER INPUT	Immediate	1	Indicates that the switch module does not wait for a trigger before starting the next entry in the scan.
	External	2	Means the trigger is going out to an external device through a trigger output connection.
	Software Trigger Function	3	The trigger will be a software command sent by the program though the IviSwtch Send Software Trigger operation.
	TTL0	111	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG0.
	TTL1	112	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG1.
	TTL2	113	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG2.
	TTL3	114	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG3.
	TTL4	115	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG4.
	TTL5	116	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG5.
	TTL6	117	A backplane TTL trigger, such as is

			found in the VXIbus or PXI TRIG6.
	TTL7	118	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG7.
	ECL0	119	A backplane ECL trigger, such as is found in the VXIbus.
	ECL1	120	A backplane ECL trigger, such as is found in the VXIbus.
	PXI Star	125	A PXI Star trigger bus.
	RTSI0	140	The RTSI 0 trigger line usually found in DAQ instruments.
	RTSI1	141	The RTSI 1 trigger line usually found in DAQ instruments.
	RTSI2	142	The RTSI 2 trigger line usually found in DAQ instruments.
	RTSI3	143	The RTSI 3 trigger line usually found in DAQ instruments.
	RTSI4	144	The RTSI 4 trigger line usually found in DAQ instruments.
	RTSI5	145	The RTSI 5 trigger line usually found in DAQ instruments.
	RTSI6	146	The RTSI 6 trigger line usually found in DAQ instruments.
SCAN ADVANCED OUTPUT	None	0	No scan advanced output trigger is sent out of the switch module.
	External	2	Means the trigger is going out to an external device through a trigger output connection.
	GPIB Service Request	5	The scan advanced output trigger is represented as a GPIB SRQ event.
	TTL0	111	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG0.
	TTL1	112	A backplane TTL trigger, such as is

			found in the VXIbus or PXI TRIG1.
	TTL2	113	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG2.
	TTL3	114	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG3.
	TTL4	115	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG4.
	TTL5	116	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG5.
	TTL6	117	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG6.
	TTL7	118	A backplane TTL trigger, such as is found in the VXIbus or PXI TRIG7.
	ECL0	119	A backplane ECL trigger, such as is found in the VXIbus.
	ECL1	120	A backplane ECL trigger, such as is found in the VXIbus.
	PXI Star	125	A PXI Star trigger bus.
	RTSI0	140	The RTSI 0 trigger line usually found in DAQ instruments.
	RTSI1	141	The RTSI 1 trigger line usually found in DAQ instruments.
	RTSI2	142	The RTSI 2 trigger line usually found in DAQ instruments.
	RTSI3	143	The RTSI 3 trigger line usually found in DAQ instruments.
	RTSI4	144	The RTSI 4 trigger line usually found in DAQ instruments.
	RTSI5	145	The RTSI 5 trigger line usually found in DAQ instruments.
	RTSI6	146	The RTSI 6 trigger line usually found in DAQ instruments.

Related Topics

IviSwtch VI Tree

IviSwtch Properties

IviSwtch Error and Warning Codes

The VIs in this class driver generate return values that report the status of the operation.

Status Code Ranges

Status Code Type	Numeric Range (in Hex)
IviSwtch Errors	0xBFFA2001 to 0xBFFA3FFF
IviSwtch Warnings (Scroll down to see these values.)	0x3FFA2001 to 0x3FFA3FFF
IVI Specific Driver Errors	0xBFFA4000 to 0xBFFA5FFF
IVI Specific Driver Warnings	0x3FFA4000 to 0x3FFA5FFF
IVI Errors	BFFA0000 to BFFA1FFF
IVI Warnings	3FFA0000 to 3FFA1FFF
Common Instrument Driver Errors	BFFC0000 to BFFCFFFF
Common Instrument Driver Warnings	3FFC0000 to 3FFCFFFF
VISA Errors	BFFF0000 to BFFFFFFF
VISA Warnings	3FFF0000 to 3FFFFFFF

IviSwTch Error Codes

Error	Value
IVISWTCH_ERROR_INVALID_SWITCH_PATH	0xBFFA2001
IVISWTCH_ERROR_INVALID_SCAN_LIST	0xBFFA2002
IVISWTCH_ERROR_RSRC_IN_USE	0xBFFA2003
IVISWTCH_ERROR_EMPTY_SCAN_LIST	0xBFFA2004
IVISWTCH_ERROR_EMPTY_SWITCH_PATH	0xBFFA2005
IVISWTCH_ERROR_SCAN_IN_PROGRESS	0xBFFA2006
IVISWTCH_ERROR_NO_SCAN_IN_PROGRESS	0xBFFA2007

IVISWTCH_ERROR_NO_SUCH_PATH	0xBFFA2008
IVISWTCH_ERROR_IS_CONFIGURATION_CHANNEL	0xBFFA2009
IVISWTCH_ERROR_NOT_A_CONFIGURATION_CHANNEL	0xBFFA200A
IVISWTCH_ERROR_ATTEMPT_TO_CONNECT_SOURCES	0xBFFA200B
IVISWTCH_ERROR_EXPLICIT_CONNECTION_EXISTS	0xBFFA200C

IVISWTCH_ERROR_LEG_MISSING_FIRST_CHANNEL	0xBFFA200D
IVISWTCH_ERROR_LEG_MISSING_SECOND_CHANNEL	0xBFFA200E
IVISWTCH_ERROR_CHANNEL_DUPLICATED_IN_LEG	0xBFFA200F
IVISWTCH_ERROR_CHANNEL_DUPLICATED_IN_PATH	0xBFFA2010
IVISWTCH_ERROR_PATH_NOT_FOUND	0xBFFA2011
IVISWTCH_ERROR_DISCONTINUOUS_PATH	0xBFFA2012
IVISWTCH_ERROR_CANNOT_CONNECT_DIRECTLY	0xBFFA2013

IVISWTCH_ERROR_CHANNELS_ALREADY_CONNECTED	0xBFFA2014
IVISWTCH_ERROR_CANNOT_CONNECT_TO_ITSELF	0xBFFA2015
IVISWTCH_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001

IviSwtch Warning Codes

Warning	Value	Message
IVISWTCH_WARN_PATH_REMAINS	0x3FFA2001	Some connections remain disconnected
IVISWTCH_WARN_IMPLICIT_CONNECTION_EXISTS	0x3FFA2002	The connections are in a connected state

Related Topic

[IVI Status Codes](#)

IviSwtch VIs

Expand this book to view an alphabetized list of IviSwtch VIs.

IviSwtch Abort Scan

IviSwtchScanner Capability Group [SCN]

This VI aborts a previously initiated scan. You initiate a scan with the [IviSwtch Initiate Scan \[SCN\]](#) VI.

If the instrument is not currently scanning, this VI returns the `IVISWTCH_ERROR_NO_SCAN_IN_PROGRESS` (0xBFFA2007) error.

You can determine the current status of the scan by calling the [IviSwtch Is Scanning](#) VI.



 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

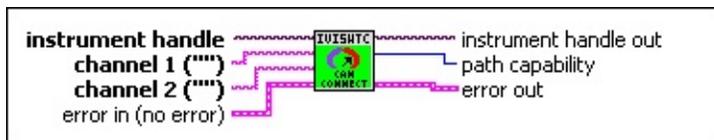
IviSwtch Can Connect Channels

IviSwtch Base Capability Group

This VI verifies that the switch module is capable of creating a path between the two channels you specify with the **channel 1** and **channel 2** parameters. If the switch module is capable of creating a path, this VI indicates whether the path is currently available given the existing connections.

If the path is not available due to currently existing connections, but the implicit connection between the two channels already exists, the VI returns the warning

IVISWTCH_WARN_IMPLICIT_CONNECTION_EXISTS (0x3FFA2002).



I/O **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

abc **channel 1 (\"'\")** You identify a path with two channels. Pass one of the virtual channel names for which you want to verify a path. Pass the other channel name as the **channel 2** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: \"'\")

channel 2 (\"'\") You identify a path with two channels. Pass one of the virtual channel names for which you want to verify a path. Pass the other channel name as the **channel 1** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwTch Initialize](#) or [IviSwTch Initialize With Options](#) VI.

error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

path capability Indicates whether a path is valid. Possible values include:

IVISWTCH_VAL_PATH_AVAILABLE1

IVISWTCH_VAL_PATH_EXISTS2

IVISWTCH_VAL_PATH_UNSUPPORTED3

IVISWTCH_VAL_RSRC_IN_USE4

IVISWTCH_VAL_SOURCE_CONFLICT5

IVISWTCH_VAL_CHANNEL_NOT_AVAILABLE6

 **Note**

1. IVISWTCH_VAL_PATH_AVAILABLE indicates that the driver

can create the path at this time.

2. `IVISWTCH_VAL_PATH_EXISTS` indicates that the explicit path between the channels already exists.
3. `IVISWTCH_VAL_PATH_UNSUPPORTED` indicates that the instrument is not capable of creating a path between the two channels.
4. `IVISWTCH_VAL_RSRC_IN_USE` indicates that although the path is valid, the driver cannot create the path at this moment because the switch module is currently using one or more of the required channels to create another path. You must destroy the other path before creating this one.
5. `IVISWTCH_VAL_SOURCE_CONFLICT` indicates that the instrument cannot create a path because both channels are connected to a different source channel.
6. `IVISWTCH_VAL_CHANNEL_NOT_AVAILABLE` indicates that the driver cannot create a path between the two channels because one of the channels is a configuration channel and thus unavailable for external connections.

IviSwtch Configure Scan List

IviSwtchScanner Capability Group [SCN]

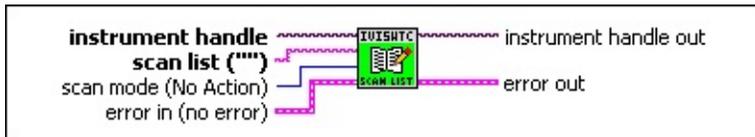
This VI configures the switch module for scanning.

Use the [IviSwtch Configure Scan Trigger](#) VI to configure the scan trigger.
Use the [IviSwtch Initiate Scan](#) VI to start the scan.

If the switch module is currently scanning through the scan list, this VI returns the `IVISWTCH_ERROR_SCAN_IN_PROGRESS` (0xBFFA2006) error without configuring the scan list.

If the given scan list string contains incorrect syntax, this VI returns the `IVISWTCH_ERROR_INVALID_SCAN_LIST` (0xBFFA2002) error.

If the given scan list string is empty, this VI returns the `IVISWTCH_ERROR_EMPTY_SCAN_LIST` (0xBFFA2004) error.



path capability Indicates whether a path is valid. Possible values include:

`IVISWTCH_VAL_PATH_AVAILABLE1`

`IVISWTCH_VAL_PATH_EXISTS2`

`IVISWTCH_VAL_PATH_UNSUPPORTED3`

`IVISWTCH_VAL_RSRC_IN_USE4`

`IVISWTCH_VAL_SOURCE_CONFLICT5`

`IVISWTCH_VAL_CHANNEL_NOT_AVAILABLE6`



Notes

1. `IVISWTCH_VAL_PATH_AVAILABLE` indicates that the driver can create the path at this time.
2. `IVISWTCH_VAL_PATH_EXISTS` indicates that the explicit path between the channels already exists.
3. `IVISWTCH_VAL_PATH_UNSUPPORTED` indicates that the

instrument is not capable of creating a path between the two channels.

4. `IVISWTCH_VAL_RSRC_IN_USE` indicates that although the path is valid, the driver cannot create the path at this moment because the switch module is currently using one or more of the required channels to create another path. You must destroy the other path before creating this one.
5. `IVISWTCH_VAL_SOURCE_CONFLICT` indicates that the instrument cannot create a path because both channels are connected to a different source channel.
6. `IVISWTCH_VAL_CHANNEL_NOT_AVAILABLE` indicates that the driver cannot create a path between the two channels because one of the channels is a configuration channel and thus unavailable for external connections.

 **instrument handle** The instrument handle that you obtain from the [IviSwTch Initialize](#) or [IviSwTch Initialize With Options](#) VI.

scan list ("") Pass the scan list you want the instrument to use. The driver uses this value to set the [IviSwTch Scan List \[SCN\]](#) property.

The scan list is a string that specifies channel connections and trigger conditions for scanning. After you call the [IviSwTch Initiate Scan](#) VI, the instrument makes or breaks connections and waits for triggers according to the instructions in the scan list.

The scan list is comprised of channel names that you separate with special characters. These special characters determine the operation the scanner performs on the channels when it executes this scan list. See the [IviSwTch Scan List](#) property for detailed description on the scan list syntax.

scan mode (No Action) Pass the scan mode you want the instrument to use. The driver sets the [IviSwTch Scan Mode \[SCN\]](#) property to this value.

The scan mode specifies how the instrument breaks existing connections when scanning.

Defined values:

`IVISWTCH_VAL_NONE (0)`—The instrument takes no action on

existing connections.

IVISWTCH_VAL_BREAK_BEFORE_MAKE (1)—The instrument breaks an existing connection before creating a new one. This is useful if you want to prevent two channels from connecting together in the transitional period.

IVISWTCH_VAL_BREAK_AFTER_MAKE (2)—The instrument breaks an existing connection after creating a new one. This is useful if you want to prevent damage from occurring on inductive elements of a circuit that cannot stand rapid changes in the current flow.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

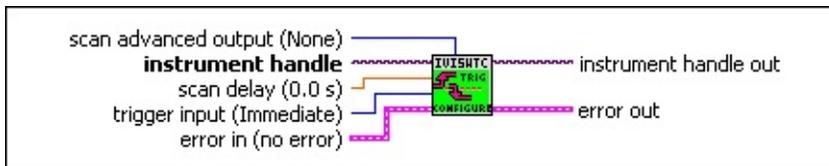
The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Configure Scan Trigger

IviSwtchScanner Capability Group [SCN]

This VI configures the scan triggers for the scan list you establish with [IviSwtch Configure Scan List](#) VI.

If the switch module is currently scanning through the scan list, this VI returns the IVISWTCH_ERROR_SCAN_IN_PROGRESS (0xBFFA2006) error without configuring the scan trigger.



I/O **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

scan delay (0.0 s) Pass the minimum length of time you want the instrument to wait from the time the instrument creates a path until it asserts a trigger on the Scan Advanced output line. Express this value in seconds. The driver uses this value to set the [IviSwtch Scan Delay \[SCN\]](#) property.

trigger input (Immediate) Pass the trigger source you want the instrument to use for scanning. The driver uses this value to set the [IviSwtch Trigger Input \[SCN\]](#) property.

The switch module waits for the trigger you specify when it encounters a semicolon in the scan list. When the trigger occurs, the switch module advances to the next entry in the scan list.

[Defined Values](#)

scan advanced output (None) Pass the output on which you want the instrument to assert the scan advanced signal. The driver uses this value to set the [IviSwtch Scan Advanced Output \[SCN\]](#) property.

After the instrument processes each entry in the scan list, it waits the length of time you specify in the **Scan Delay** parameter and then asserts a trigger on the line you specify with this parameter.

Defined Values



error in (no error) The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.



error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Connect Channels

IviSwtch Base Capability Group

This VI creates a path between channel 1 and channel 2. The driver calculates the shortest path between the two channels. [Details](#)

This VI applies default values to properties that have not been set by you under the following conditions:

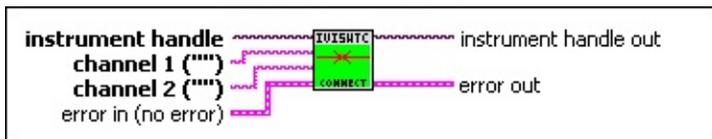
If you have not set the value of any property in the IviSwtchScanner extension, the following default values are used:

[IviSwtch Scan List \[SCN\]](#)"" (Empty string)

[IviSwtch Trigger Input \[SCN\]](#)IVISWTCH_VAL_EXTERNAL

[IviSwtch Scan Advanced Output \[SCN\]](#)IVISWTCH_VAL_EXTERNAL

[IviSwtch Scan Delay \[SCN\]](#)0



instrument handle The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

channel 1 (\"'\") You identify a path with two channels. Pass one of the virtual channel names for which you want to create a path. Pass the other channel name as the **channel 2** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

channel 2 (\"'\") You identify a path with two channels. Pass one of

the virtual channel names for which you want to create a path. Pass the other channel name as the **channel 1** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel in MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

If a path is not available, the VI returns one of the following errors:

IVISWTCH_ERROR_EXPLICIT_CONNECTION_EXISTS

(0xBFFA200C)if the two channels are already explicitly connected by calling either the IviSwtch Connect Channels or IviSwtch Set Path VI.

IVISWTCH_ERROR_IS_CONFIGURATION_CHANNEL (0xBFFA2009)if a channel is a configuration channel. Call IviSwtch_GetError for information about which of the two channels is a configuration channel.

IVISWTCH_ERROR_ATTEMPT_TO_CONNECT_SOURCES

(0xBFFA200B)if both channels are connected to a different source. Call IviSwtch_GetError for sources to which channel 1 and 2 connect.

IVISWTCH_ERROR_CANNOT_CONNECT_TO_ITSELF

(0xBFFA2015)if channels 1 and 2 are one and the same channel.

IVISWTCH_ERROR_PATH_NOT_FOUND (0xBFFA2011) if the driver cannot find a path between the two channels.

IviSwtch Connect Channels Details



Notes

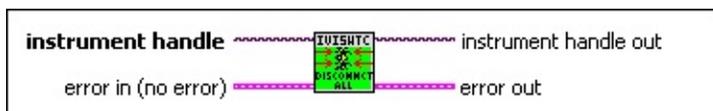
1. The paths are bidirectional. For example, if a path exists from channel CH1 to CH2, then a path from channel CH2 to CH1 also exists.
2. This VI performs interchangeability checking when the [IviSwtch Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviSwtch Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the help text for the IviSwtch Interchange Check property.

IviSwtch Disconnect All Channels

IviSwtch Base Capability Group

This VI disconnects all paths created since the [IviSwtch Initialize](#) VI or [IviSwtch Reset](#) VI has been called.

 **Note** If the switch module is not capable of disconnecting all paths, this VI returns the warning IVISWTCH_WARN_PATH_REMAINS (0x3FFA2001).



 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Disconnect Channels

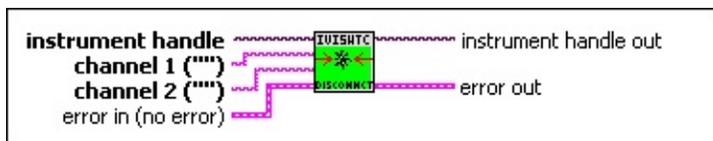
IviSwtch Base Capability Group

This VI destroys the path between two channels that you create with the [IviSwtch Connect Channels](#) or [IviSwtch Set Path](#) VI.

This VI returns as soon as the command is given to the switch module and the switch module is ready for another command. This may be before or after the switches involved settle. Use the [IviSwtch Switch Is Debounced](#) VI to see if the switch has settled. Use the [IviSwtch Wait For Debounce](#) VI if you want to wait until the switch has debounced.

If some connections remain after disconnecting the two specified channels, this VI returns the warning IVISWTCH_WARN_PATH_REMAINS (0x3FFA2001) error.

If no explicit path exists between the two specified channels, this VI returns the IVISWTCH_ERROR_NO_SUCH_PATH (0xBFFA2008) error without performing any disconnection operation.



instrument handle The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

channel 1 (\"") You identify a path with two channels. Pass one of the virtual channel names for which you want to destroy a path. Pass the other channel name as the **channel 2** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

- channel 2 ("")** You identify a path with two channels. Pass one of the virtual channel names for which you want to destroy a path. Pass the other channel name as the **channel 1** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

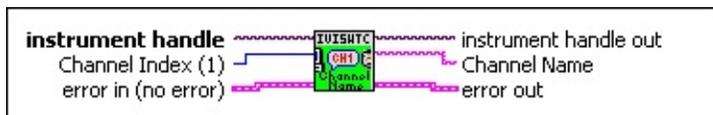
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtchGet Channel Name

IviSwtch Base Capability Group

This VI returns the specific driver defined channel name that corresponds to the one-based index you specify. If the value you pass for the **Index** parameter is less than one or greater than the value of the Channel Count property, the VI returns an empty string in the **Name** parameter and returns an error.



- instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI. The handle identifies a particular instrument session.
- index (1)** A one-based index that defines which name to return. The index must be less than or equal to the [ClassPrefix Channel Count](#) property.
- instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI. The handle identifies a particular instrument session.
- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.
- output channel name** Returns the name of the Channel specified by the **index** parameter.
- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Get Path

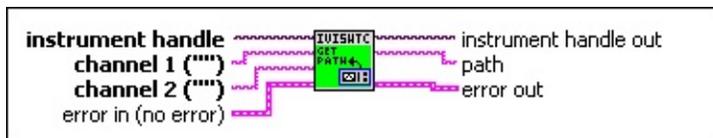
IviSwtch Base Capability Group

In some cases there is more than one possible path between two channels. The driver or the instrument selects the path when you connect two channels with the [IviSwtch Connect Channels](#) VI. Thus, you cannot guarantee that every call to the IviSwtch Connect Channels VI establishes exactly the same path when you pass the same channels. This VI returns a string that uniquely identifies the path you create with the IviSwtch Connect Channels VI. You can pass this string to the [IviSwtch Set Path](#) VI to establish the exact same path in the future.

If no explicit path exists between the two specified channels, this VI returns the IVISWTCH_ERROR_NO_SUCH_PATH (0xBFFA2008) error.



Note This VI returns only those paths that you explicitly create by calling IviSwtch Connect Channels and IviSwtch Set Path VIs. For example, if you connect channels CH1 and CH3, and then channels CH2 and CH3, the explicit path between channels CH1 and Ch2 does not exist and this VI returns the IVISWTCH_ERROR_NO_SUCH_PATH (0xBFFA2008) error.



- instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.
- channel 1 (\"'\")** You identify a path with two channels. Pass one of the virtual channel names for which you want to get a path. Pass the other channel name as the **channel 2** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an

instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

- channel 2 ("")** You identify a path with two channels. Pass one of the virtual channel names for which you want to get a path. Pass the other channel name as the **channel 1** parameter.

Virtual channel names are aliases for instrument specific channel strings. The instrument specific channel strings can differ from one instrument to another. Virtual channel names allow you to use and swap instruments without having to change the channel names in your source code. You assign a virtual channel name to an instrument specific channel through MAX. This control accepts virtual channel names you have assigned to the specific instrument you are using. It also accepts the instrument specific channel names.

Default Value: ""

-  **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

-  **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.
- path** The comma-separated path between channels you specify in the **Channel 1** and **Channel 2** parameters.

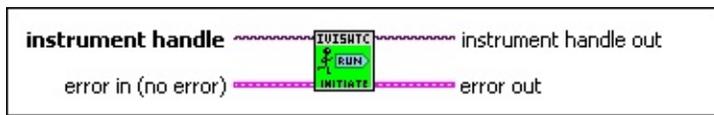
-  **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Initiate Scan

IviSwtchScanner Capability Group [SCN]

This VI initiates a scan using the scan list and triggers you configure with the [IviSwtch Configure Scan List](#) and [IviSwtch Configure Scan Trigger \[SCN\]](#) VIs. If there is no scan list, this VI returns the error IVISWTCH_ERROR_EMPTY_SCAN_LIST (0xBFFA2004). This VI returns immediately.



Once you start the scanning operation, you cannot perform any other operation other than Get Property, Abort Scan, or Send Software Trigger. All other VIs return the IVISWTCH_ERROR_SCAN_IN_PROGRESS (0xBFFA2006) error.

To stop the scanning operation, call the [IviSwtch Abort Scan](#) VI.

This VI applies default values to properties that have not been set by you under the following conditions:

If you have not set the value of any property in the IviSwtchScanner extension, the following default values are used:

[IviSwtch Scan List \[SCN\]](#)—"" (Empty string)

[IviSwtch Trigger Input \[SCN\]](#)—IVISWTCH_VAL_EXTERNAL

[IviSwtch Scan Advanced Output \[SCN\]](#)—IVISWTCH_VAL_EXTERNAL

[IviSwtch Scan Delay \[SCN\]](#)—0



Notes

1. The paths are bidirectional. For example, if a path exists from channel CH1 to CH2, then a path from channel CH2 to CH1 also exists.
2. This VI performs interchangeability checking when the [IviSwtch Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviSwtch Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about

interchangeability checking, refer to the [IviSwtch Interchange Check](#) property.

 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Send Software Trigger

IviSwtchSoftwareTrigger Capability Group [SWT]

This VI sends a command to trigger the instrument. You must satisfy certain conditions before you can successfully call this VI. The [IviSwtch Trigger Input](#) property is set to IVISWTCH_VAL_SOFTWARE_TRIG.

If this condition is not true, this VI does not send the software trigger and returns the error IVISWTCH_ERROR_TRIGGER_NOT_SOFTWARE.

This VI does not check the instrument status. Typically, the end-user calls this VI only in a sequence of calls to other low-level driver VIs. The sequence performs one operation. The end-user uses the low-level VIs to optimize one or more aspects of interaction with the instrument. To check the instrument status, call the IviSwtch Error-Query VI at the conclusion of the sequence.



 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

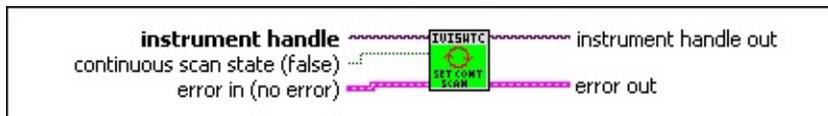
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Set Continuous Scan

IviSwtchScanner Capability Group [SCN]

This VI sets the continuous scan mode on the instrument.



I/O **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

continuous scan state (false) Specifies whether the continuous scan mode is enabled or disabled.

FALSE—disabled

TRUE—enabled

E+I **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

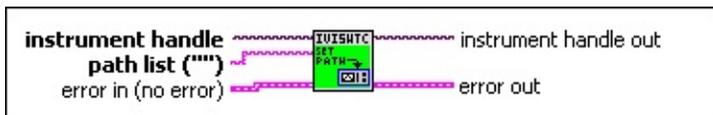
E+I **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Set Path

IviSwtch Base Capability Group

This VI connects two channels by establishing the exact path you specify with the **Path List** parameter. This VI allows you to bypass the automatic routing feature provided by the IviSwtch driver, and gives you deterministic control over the path that is created between two channels. You can specify the exact path, in terms of configuration channels, used to connect two channels. The first and last entries in the pathList represent the two channels you want to connect.



The driver makes a connection between the channels using the configuration channels. These intermediary steps are called legs of the path. The format of the leg of the path is `ch1->conf1`, where `ch1` and `conf1` are the two channels the driver used to establish the connection between the first and the last channel. The path list syntax is a comma-separated list of path legs that obey the following rules:

- The second channel of a leg in the path list must be the same as the first channel in the subsequent leg.
- Every channel in the path list other than the first and the last must be a configuration channel.

An example of a path list is: `ch1»conf1,conf1»ch2`

This string is *not* interchangeable because the names of switches within the switch module are not required to be interchangeable and depend on the internal architecture of the switch module. However, you can use the [IviSwtch Connect Channels](#) and then [IviSwtch Get Path](#) VIs to retrieve an already existing path. This allows you to guarantee that the routing can be recreated exactly.

If the instrument cannot parse the input path list string, this VI returns the error `IVISWTC_ERROR_INVALID_SWITCH_PATH (0xBFFA2001)`.

If the specified path list string is empty, this VI returns the error

IVISWTCH_ERROR_EMPTY_SWITCH_PATH (0xBFFA2005).

If one of the channels in the path list is a configuration channel that is currently in use, this VI returns the error
IVISWTCH_ERROR_RSRC_IN_USE (0xBFFA2003).

If an explicit connection is made to a configuration channel, this VI returns the error
IVISWTCH_ERROR_IS_CONFIGURATION_CHANNEL (0xBFFA2009).

If one of the non-terminal channels in the path list is not a configuration channel, this VI returns the error
IVISWTCH_ERROR_NOT_A_CONFIGURATION_CHANNEL (0xBFFA200A).

If the path list attempts to connect between two different source channels, this VI returns the error
IVISWTCH_ERROR_ATTEMPT_TO_CONNECT_SOURCES (0xBFFA200B).

If the path list attempts to connect between channels that already have an explicit connection, this VI returns the error
IVISWTCH_ERROR_EXPLICIT_CONNECTION_EXISTS (0xBFFA200C).

If a leg in the path list does not begin with a channel name, this VI returns the error
IVISWTCH_ERROR_LEG_MISSING_FIRST_CHANNEL (0xBFFA200D).

If a leg in the path list is missing the second channel, this VI returns the error
IVISWTCH_ERROR_LEG_MISSING_SECOND_CHANNEL (0xBFFA200E).

If the first and the second channels in the leg are the same, this VI returns the error
IVISWTCH_ERROR_CHANNEL_DUPLICATED_IN_LEG (0xBFFA200F).

If a channel name is duplicated in the path string, this VI returns the error
IVISWTCH_ERROR_CHANNEL_DUPLICATED_IN_PATH (0xBFFA2010).

If the first channel of a leg in the path is not the same as the second channel in the previous leg, this VI returns the error
IVISWTCH_ERROR_DISCONTINUOUS_PATH (0xBFFA2012).

If the path list contains a leg with two channels that cannot be directly

connected, this VI returns the error `IVISWTCH_ERROR_CANNOT_CONNECT_DIRECTLY` (0xBFFA2013).

If a leg in the path contains two channels that are already directly connected, this VI returns the error `IVISWTCH_ERROR_CHANNELS_ALREADY_CONNECTED` (0xBFFA2014).

This VI applies default values to properties that are not set by you under the following condition:

(1) If the value of any property is not set in the `IviSwchScanner` extension, the following default values are used:

[IviSwch Scan List \[SCN\]](#)—"" (Empty string)

[IviSwch Trigger Input \[SCN\]](#)—`IVISWTCH_VAL_EXTERNAL`

[IviSwch Scan Advanced Output \[SCN\]](#)—`IVISWTCH_VAL_EXTERNAL`

[IviSwch Scan Delay \[SCN\]](#)—0



Notes

1. The paths are bidirectional. For example, if a path exists from CH1 to CH2, then a path from CH2 to CH1 also exists.
2. This VI performs interchangeability checking when the [IviSwch Interchange Check](#) property is set to TRUE. Use NI Spy to view interchangeability warnings. Alternately, you may use the [IviSwch Get Next Interchange Warning](#) VI to retrieve interchangeability warnings. For more information about interchangeability checking, refer to the [IviSwch Interchange Check](#) property.



instrument handle The instrument handle that you obtain from the [IviSwch Initialize](#) or [IviSwch Initialize With Options](#) VI.



path list ("") Pass the path list for the path you previously created that you want the switch module to establish. You obtain the path list for a path you previously created with the `IviSwch Get Path` VI.

Example: The path from Ch1 to Ch3 over Conf1 is:

"Ch1»Conf1,Conf1»Ch3"



error in (no error) The error in cluster can accept error information

wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.



instrument handle out The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.



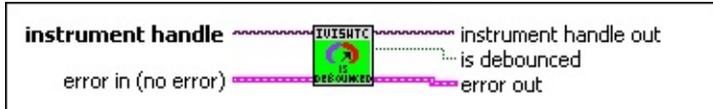
error out The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Switch Is Debounced

IviSwtch Base Capability Group

This VI returns a value that indicates whether all the paths that you previously created have settled.



 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

is debounced Indicates the state of the switch module. The driver returns the value of the [IviSwtch Is Debounced](#) property.

TRUE indicates that all the paths that you created have settled.

FALSE indicates that all the paths that you created have not settled.

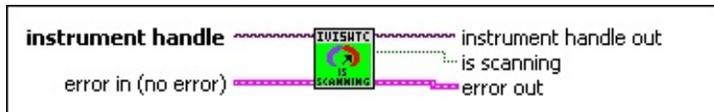
 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Switch Is Scanning

IviSwtchScanner Capability Group [SCN]

This VI returns the state of the switch module. It indicates if the instrument is currently scanning or is idle.



I/O **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

E **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

is scanning Indicates the state of the switch module. The driver returns the value of [IviSwtch Is Scanning \[SCN\]](#) property.

TRUE indicates that the switch module is scanning.

FALSE indicates that the switch module is idle.

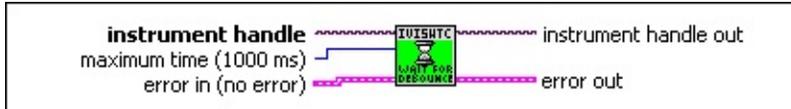
E **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Wait For Debounce

IviSwtch Base Capability Group

Calling this VI causes the driver to return process control back to you only after all the paths that you previously created have settled.



I/O **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

I32 **maximum time (1000 ms)** Specifies the maximum length of time for this VI to wait until all switches in the switch module debounce.

If the time you specify elapses before all switches debounce, this VI returns a timeout error.

The units are milliseconds.

Default Value:1000 (ms)

 **Note** The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.

E51 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

I/O **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

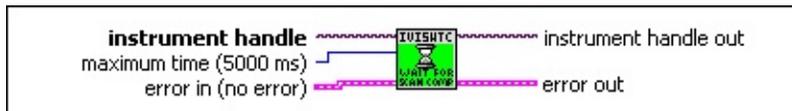
E51 **error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Wait For Scan To Complete

IviSwtchScanner Capability Group [SCN]

This VI waits until the instrument stops scanning or the time you specified has elapsed. If the switch module is not currently scanning through the scan list, this VI returns the IVISWTCH_ERROR_NO_SCAN_IN_PROGRESS (0xBFFA2007) error. If the time you specified elapses before the instrument stops scanning, this VI returns an error.



 **instrument handle** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

maximum time (1000 ms) Specifies the maximum length of time for this VI to wait until the instrument stops scanning.

If the time you specify elapses before it stops scanning, this VI returns a timeout error.

The units are milliseconds.

Default value: 1000 ms.



Note The **Maximum Time** parameter affects only this VI. It has no effect on other timeout parameters or properties.

 **error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

 **instrument handle out** The instrument handle that you obtain from the [IviSwtch Initialize](#) or [IviSwtch Initialize With Options](#) VI.

 **error out** The error out cluster passes error or warning information

out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

IviSwtch Properties

Expand this book to view an alphabetized list of IviSwtch properties.

IviSwtch Maximum AC Voltage

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum AC voltage the channel can switch. The units are volts RMS.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum AC Voltage

IviSwtch Bandwidth

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the bandwidth for the channel. The units are hertz.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Bandwidth

IviSwtch Maximum Carry AC Current

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum AC current the channel can carry. The units are amperes RMS.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Carry AC Current

IviSwtch Maximum Carry AC Power

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum AC power the channel can carry. The units are volt-amperes.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Carry AC Power

IviSwtch Maximum Carry DC Current

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum DC current the channel can carry. The units are amperes.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Carry DC Current

IviSwtch Maximum Carry DC Power

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum DC power the channel can switch. The units are watts.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Carry DC Power

IviSwtch Channel Count

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of channel available on the instrument.

Property Node Path

IviSwtch&»Inherent IVI Settings»Specific Driver Capabilities»Channel Count

IviSwtch Characteristic Impedance

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the characteristic impedance for the channel. The units are ohms.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel

Property Node Path

IviSwtch»Module Characteristics»Characteristic Impedance

IviSwtch Is Configuration Channel

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	None

Description

This channel-based property specifies whether to reserve the channel for internal path creation. A channel that is available for internal path creation is called a configuration channel. The driver may use configuration channels to create paths between two channels you specify in the [IviSwtch Connect Channels VI](#). Configuration channels are not available for external connections.

Set this property to TRUE to mark the channel as a configuration channel. Set this property to FALSE to mark the channel as available for external connections.

After you identify a channel as a configuration channel, you cannot use that channel for external connections. The IviSwtch Connect Channels VI returns the IVISWTCH_ERROR_IS_CONFIGURATION_CHANNEL error when you attempt to establish a connection between a configuration channel and any other channel.

Property Node Path

IviSwtch»Channel Configuration»Is Configuration Channel

IviSwtch Continuous Scan [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	N/A	IviSwtch Set Continuous Scan

Description

This property specifies whether the switch module continues scanning from the top of the scan list after reaching the end of the list. A value of TRUE indicates that the switch module continuously scans. A value of FALSE indicates that the switch module does not continuously scan.

If you set this property to TRUE, the VI [IviSwtch Wait For Scan To Complete](#) always times out, and you must call [IviSwtch Abort Scan](#) to stop the scan.

Property Node Path

IviSwtch»Scanning Configuration»Continuous Scan

IviSwtch Maximum DC Voltage

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum DC voltage the channel can switch. The units are volts.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum DC Voltage

IviSwtch Is Debounced

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviSwtch Switch Is Debounced

Description

This property indicates whether the entire switch module has settled since the last switching command. A value of TRUE indicates that all signals going through the switch module are valid.

Property Node Path

IviSwtch»Module Characteristics»Is Debounced

IviSwtch Number of Columns [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	None

Description

This property returns the number of columns of a matrix or scanner. If the switch module is a scanner, this value is the number of input channels.

The [IviSwtch Wire Mode](#) property affects the number of available columns. For example, if your module has 8 input lines and you use the 4-wire mode, then the number of columns you have available is 2.

Property Node Path

IviSwtch»Matrix Configuration»Number of Columns [SCN]

IviSwtch Number of Rows [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	None

Description

This property returns the number of rows of a matrix or scanner. If the switch module is a scanner, this value is the number of output channels.

The [IviSwtch Wire Mode](#) property affects the number of available rows. For example, if your module has 2 output lines and you use the 2-wire mode, then the number of rows you have available is 1.

Property Node Path

IviSwtch»Matrix Configuration»Number of Rows [SCN]

IviSwtch Scan Advanced Output [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviSwtch Configure Scan Trigger

Description

This property specifies the method you want to use to notify another instrument that all signals going through the switch module have settled following the processing of one entry in the scan list.

This trigger is asserted each time a path is created. This trigger should not be asserted until after sufficient settling time has been given for the path. The [IVISWTCH_VAL_EXTERNAL](#) value means the trigger is going out to an external device through a trigger output connection.

Property Node Path

IviSwtch»Scanning Configuration»Scan Advanced Output [SCN]

Defined Values

[IVISWTCH_VAL_NONE](#)

[IVISWTCH_VAL_EXTERNAL](#)

[IVISWTCH_VAL_GPIB_SRQ](#)

[IVISWTCH_VAL_TTL0](#)

[IVISWTCH_VAL_TTL1](#)

[IVISWTCH_VAL_TTL2](#)

[IVISWTCH_VAL_TTL3](#)

[IVISWTCH_VAL_TTL4](#)

[IVISWTCH_VAL_TTL5](#)

[IVISWTCH_VAL_TTL6](#)

[IVISWTCH_VAL_TTL7](#)

[IVISWTCH_VAL_ECL0](#)

[IVISWTCH_VAL_ECL1](#)

IVISWTCH_VAL_PXI_STAR

IVISWTCH_VAL_RTSI_0

IVISWTCH_VAL_RTSI_1

IVISWTCH_VAL_RTSI_2

IVISWTCH_VAL_RTSI_3

IVISWTCH_VAL_RTSI_4

IVISWTCH_VAL_RTSI_5

IVISWTCH_VAL_RTSI_6

IviSwtch Scan Delay [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviSwtch Configure Scan Trigger

Description

This property specifies the minimum amount of time the switch module waits before it asserts the scan advanced output trigger after opening or closing the switch. The switch module always waits for debounce before asserting the trigger. Thus, the actual delay is always the greater value of the settling time and the value you specify as the switch delay.

The units are seconds. Due to different designs of the switch modules, the actual time might be longer.

Property Node Path

IviSwtch»Scanning Configuration»Scan Delay [SCN]

IviSwtch Scan List [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviSwtch Configure Scan List

Description

This property contains a scan list, which is a string that specifies channel connections and trigger conditions. The [IviSwtch Initiate Scan \[SCN\]](#) VI makes or breaks connections and waits for triggers according to the instructions in the scan list.

The scan list is comprised of channel names that you separate with special characters. These special characters determine the operations the scanner performs on the channels when it executes this scan list.

The list scan list contains the channels, in the order to be scanned. You can refer to the following table for scan list syntax.

Symbol	Symbol Name	Description	Syntax Examples
;	Semi-colon	This character instructs the driver to wait for an input trigger event before proceeding to the next instruction in the scan list string. In the example, the driver notifies the switch module to create a path from channel CH1 to channel CH2, wait for a trigger, and then create a path from channel CH3 to channel CH4.	CH1 -> CH2 ; CH3 -> CH4
&	Ampersand	This character instructs the driver to connect all the paths separated by the symbol at the same time, before the next trigger event. However, the driver does not guarantee the order of connection, except that all connections are settled before the next trigger event. In the example, the driver notifies the switch module to create a	CH1 -> CH2 & CH3 -> CH4 ; A -> B

		path between channels CH1 and CH2 and between channels CH3 and CH4, not necessarily in that order. The switch module then waits for a trigger before connecting channel A to channel B.	
~	Tilde	This character instructs the driver to disconnect a path. In the example, the driver notifies the switch module to disconnect channel CH1 from channel CH2. Notice that only a path connection event generate a scan-advanced trigger. Disconnecting a path will not generate a scan-advanced trigger.	~CH1 -> CH2
->	Channel Pair	This symbol signifies a channel pair, which instructs the driver to create a path between the two channels separated by the symbol. In the example, the driver notifies the switch module to create a path from channel CH1 to channel CH2.	CH1 -> CH2

Property Node Path

IviSwtch»Scanning Configuration»Scan List [SCN]

IviSwtch Scan Mode [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviSwtch Configure Scan List

Description

This property specifies what happens to existing connections that conflict with the connections you make in a scan list. For example, if CH1 is already connected to CH2 and the scan list instructs the switch module to connect CH1 to CH3, this property specifies what happens to the connection between CH1 and CH2.

If the value of this property is `IVISWTCH_VAL_NONE`, the switch module takes no action on existing paths. If the value is `IVISWTCH_VAL_BREAK_BEFORE_MAKE`, the switch module breaks conflicting paths before making new ones. If the value is `IVISWTCH_VAL_BREAK_AFTER_MAKE`, the switch module breaks conflicting paths after making new ones.

Most switch modules support only one of the possible values. In such cases, this property serves as an indicator of the module's behavior.

[IVISWTCH_VAL_BREAK_BEFORE_MAKE](#)—tells the driver to break the previous paths before making the new paths.

[IVISWTCH_VAL_BREAK_AFTER_MAKE](#)—tells the driver to make new paths before breaking the previous paths.

[IVISWTCH_VAL_NONE_for_SCAN_MODE](#)—indicates that no action should be taken on the previous paths. You must support at least one value to support this property.

Property Node Path

IviSwtch»Scanning Configuration»Scan Mode [SCN]

Defined Values

[IVISWTCH_VAL_NONE](#)

[IVISWTCH_VAL_BREAK_BEFORE_MAKE](#)

[IVISWTCH_VAL_BREAK_AFTER_MAKE](#)

IviSwtch Is Scanning [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	N/A	N/A	IviSwtch Switch Is Scanning

Description

This property indicates whether the switch module has completed the scan operation. The value TRUE indicates that the scan is complete.

Property Node Path

IviSwtch»Scanning Configuration»Is Scanning [SCN]

IviSwtch Settling Time

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum length of time from after you make a connection until the signal flowing through the channel settles. The units are seconds.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Settling Time

IviSwtch Is Source Channel

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	Channel	None	None

Description

This channel-based property specifies whether you want to identify the channel as a source channel. Typically, you set this property to TRUE when you attach the channel to a power supply, a function generator, or an active measurement point on the unit under test, and you do not want to connect the channel to another source. The driver prevents source channels from connecting to each other. The Connect Channels VI returns the IVISWTCH_ERROR_ATTEMPT_TO_CONNECT_SOURCES error when you attempt to connect two channels that you identify as source channels.

Property Node Path

IviSwtch»Channel Configuration»Is Source Channel

IviSwtch Maximum Switching AC Current

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum AC current the channel can switch. The units are amperes RMS.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Switching AC Current

IviSwtch Maximum Switching AC Power

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum AC power the channel can switch. The units are volt-amperes.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Switching AC Power

IviSwtch Maximum Switching DC Current

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum DC current the channel can carry. The units are amperes.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Switching DC Current

IviSwtch Maximum Switching DC Power

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	N/A	None

Description

This channel-based property returns the maximum DC power the channel can switch. The units are watts.

The values for this property are on per-channel basis and may not take into account the other switches that make up a path to or from this channel.

Property Node Path

IviSwtch»Module Characteristics»Maximum Switching DC Power

IviSwtch Trigger Input [SCN]

IviSwtchScanner Property [SCN]

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	R/W	N/A	None	IviSwtch Configure Scan Trigger

Description

This property specifies the source of the trigger for which the switch module can wait when processing a scan list. The switch module waits for a trigger when it encounters a semicolon in a scan list. When the trigger occurs, the switch module advances to the next entry in the scan list.

This trigger tells the switch module to advance to the next entry in the scan list and close the specified channel. The [IVISWTCH_VAL_IMMEDIATE](#) value indicates that the switch module does not wait for a trigger before starting the next entry in the scan. This is typically done for switch modules that support the [IviSwtch Scan Delay](#) property and can therefore have the switch module pace itself. The [IVISWTCH_VAL_EXTERNAL](#) value means the trigger is coming from an external source through a trigger input connection.

Property Node Path

IviSwtch»Scanning Configuration»Trigger Input [SCN]

Defined Values

[IVISWTCH_VAL_IMMEDIATE](#)

[IVISWTCH_VAL_EXTERNAL](#)

[IVISWTCH_VAL_SOFTWARE_TRIG](#)

[IVISWTCH_VAL_TTL0](#)

[IVISWTCH_VAL_TTL1](#)

[IVISWTCH_VAL_TTL2](#)

[IVISWTCH_VAL_TTL3](#)

[IVISWTCH_VAL_TTL4](#)

[IVISWTCH_VAL_TTL5](#)

IVISWTCH_VAL_TTL6

IVISWTCH_VAL_TTL7

IVISWTCH_VAL_ECL0

IVISWTCH_VAL_ECL1

IVISWTCH_VAL_PXI_STAR

IVISWTCH_VAL_RTSI_0

IVISWTCH_VAL_RTSI_1

IVISWTCH_VAL_RTSI_2

IVISWTCH_VAL_RTSI_3

IVISWTCH_VAL_RTSI_4

IVISWTCH_VAL_RTSI_5

IVISWTCH_VAL_RTSI_6

IviSwtch Wire Mode

IviSwtch Base Property

Data Type	Access	Applies to	Coercion	High Level VIs
<input type="checkbox"/>	RO	Channel	None	None

Description

This property specifies the wire mode of the switch module. This property affects the values of the [IviSwtch Number of Rows](#) and [IviSwtch Number of Columns](#) properties. The actual number of input and output lines on the switch module is fixed, but the number of channels depends on how many lines constitute each channel.

Property Node Path

IviSwtch»Matrix Configuration»Wire Mode

IviSwtch Defined Values

Expand this book to view an alphabetized list of IviSwtch defined values.

IVISWTCH_VAL_BREAK_AFTER_MAKE for IVISWTCH_ATTR_SCAN_MODE

Tells the driver to make the new paths before breaking the previous paths.

Defined Value: 2

IVISWTCH_VAL_BREAK_BEFORE_MAKE for IVISWTCH_ATTR_SCAN_MODE

Tells the devices to break the previous paths before making the new paths.

Defined Value: 1

IVISWTCH_VAL_ECL0

A backplane ECL0 trigger, such as is found in the VXIbus.

Defined Value: 119

IVISWTCH_VAL_ECL1

A backplane ECL1 trigger, such as is found in the VXIbus.

Defined Value: 120

IVISWTCH_VAL_EXTERNAL

Means the trigger is going out to an external device through a trigger output connection.

Defined Value: 2

IVISWTCH_VAL_EXTERNAL (for IVISWTCH_ATTR_SCAN_ADVANCED_OUTPUT)

The scan advanced output trigger is represented as an external trigger event

Defined Value: 2

IVISWTCH_VAL_EXTERNAL (for IVISWTCH_ATTR_TRIGGER_INPUT)

Indicates that the trigger is coming from an external source through a trigger input connection.

Defined Value: 2

IVISWTCH_VAL_GPIB_SRQ

The scan advanced output trigger is represented as a GPIB SRQ event.

Defined Value: 5

IVISWTCH_VAL_IMMEDIATE

Indicates that the switch module does not wait for a trigger before starting the next entry in the scan.

Defined Value: 1

IVISWTCH_VAL_NONE
(for_SCAN_ADVANCED_OUTPUT)

No scan advanced output trigger is sent out of the switch module.

Defined Value: 0

IVISWTCH_VAL_NONE (for_SCAN_MODE)

Indicates that no action should be taken on the previous paths.

Defined Value: 0

IVISWTCH_VAL_NONE_for_SCAN_ADVANCED_C

No scan advanced output trigger is sent out of the switch module.

Defined Value: 0

IVISWTCH_VAL_NONE for IviSwtch Scan Mode

Indicates that no action should be taken on the previous paths.

Defined Value: 0

IVISWTCH_VAL_PXI_STAR

A PXI star trigger bus.

Defined Value: 125

IVISWTCH_VAL_RTISI_0

The RTISI0 trigger line usually found in DAQ instruments.

Defined Value: 140

IVISWTCH_VAL_RTISI_1

The RTSI1 trigger line usually found in DAQ instruments.

Defined Value: 141

IVISWTCH_VAL_RTISI_2

The RTISI2 trigger line usually found in DAQ instruments.

Defined Value: 142

IVISWTCH_VAL_RTISI_3

The RTISI3 trigger line usually found in DAQ instruments.

Defined Value: 143

IVISWTCH_VAL_RTISI_4

The RTISI4 trigger line usually found in DAQ instruments.

Defined Value: 144

IVISWTCH_VAL_RTISI_5

The RTISI5 trigger line usually found in DAQ instruments.

Defined Value: 145

IVISWTCH_VAL_RTISI_6

The RTISI6 trigger line usually found in DAQ instruments.

Defined Value: 146

IVISWTCH_VAL_SOFTWARE_TRIG

The trigger is a software command sent by the program through the [IviSwtch_SendSoftwareTrigger](#) operation.

If you set the IVISWTCH_ATTR_TRIGGER_INPUT property to this value, the instrument can only be triggered by calling the IviSwtch_SendSoftwareTrigger function.

Defined Value: 3

IVISWTCH_VAL_TTL0

A backplane TTL trigger, such as the VXIbus TTL0 or PXI TRIG0.

Defined Value: 111

IVISWTCH_VAL_TTL1

A backplane TTL trigger, such as the VXIbus TTL1 or PXI TRIG1.

Defined Value: 112

IVISWTCH_VAL_TTL2

A backplane TTL trigger, such as the VXIbus TTL2 or PXI TRIG2.

Defined Value: 113

IVISWTCH_VAL_TTL3

A backplane TTL trigger, such as the VXIbus TTL3 or PXI TRIG3.

Defined Value: 114

IVISWTCH_VAL_TTL4

A backplane TTL trigger, such as the VXIbus TTL4 or PXI TRIG4.

Defined Value: 115

IVISWTCH_VAL_TTL5

A backplane TTL trigger, such as the VXIbus TTL5 or PXI TRIG5.

Defined Value: 116

IVISWTCH_VAL_TTL6

A backplane TTL trigger, such as the VXIbus TTL6 or PXI TRIG6.

Defined Value: 117

IVISWTCH_VAL_TTL7

A backplane TTL trigger, such as the VXIbus TTL7 or PXI TRIG7.

Defined Value: 118

Properties for Controlling IviSwtch Simulation



Note You can set Simulation Attributes in MAX on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** or in the simulation interactive panels. Refer to **National Instruments IVI Driver Help»Configuring Your System** for more information.

Two sets of properties exist for use with the IviSwtch Simulation Driver. The following table describes properties that control behavior of the driver. Another table lists [properties that simulate the status](#) of specific driver functions.

Properties for Controlling IviSwtch Simulation

Name	Data Type	Access	Applies to	Description
INTERACTIVE_SIMULATION	Boolean	W	N/A	Specifies whether to set the driver to interactive mode. In interactive mode, the simulation driver uses pop-up user interface panels for getting information from you. Non-interactive mode is useful when you do not want the interactive panels to interrupt your test program.
SELF_TEST_CODE	Integer	W	N/A	Specifies the self-test code. When the IviSwtch Self-Test VI is called, the self-test result parameter returns this

				value.
SELF_TEST_MSG	String	W	N/A	Specifies the self-test message. When the IviSwtch Self-Test VI is called, the self-test message parameter returns this string.
ERROR_QUERY_CODE	Integer	W	N/A	Specifies the error-query code. When the IviSwtch Error-Query VI is called, the error code parameter returns this value.
ERROR_QUERY_MSG	String	W	N/A	Specifies the error-query message. When the IviSwtch Error-Query VI is called, the error message parameter returns this string.
DRIVER_REV_QUERY	String	W	N/A	Specifies the instrument

				driver revision message. When the IviSwtch Revision Query VI is called, the instrument driver revision parameter returns this string.
INSTR_REV_QUERY	String	W	N/A	Specifies the firmware revision message. When the IviSwtch Revision Query VI is called, the firmware revision parameter returns this string.
SIMULATE_STATUS_CODE	Boolean	W	N/A	Specifies whether to simulate return codes from the IviSwtch driver VIs. Refer to the IviSwtch Properties for Status Simulation

			table.
--	--	--	------------------------

Related Topics

[Setting Up Simulated Errors](#)

[Setting Up Simulation for VXI*plug&play* VIs](#)

Properties for Status Simulation in IviSwtch

When an IviSwtch driver function is called, the driver queries whether `NISSWTCH_ATTR_SIMULATE_STATUS_CODE` is enabled. If enabled, the driver gets the appropriate function's simulation status code. If the value of the simulation status code represents a warning (has a positive value), then it is returned as the function's return status only if no other error or warning occurred before the function ends. If the value of the simulation status code represents an error (has a negative value), then it is returned as the function's return status only if no other error occurred before the function ends.

The following IviSwtch VIs do not support status code simulation:

- IviSwtch Get Error
- IviSwtch Clear Error
- IviSwtch Get Next Interchange Warning
- IviSwtch Reset Next Interchange Warning
- IviSwtch Clear Next Interchange Warning
- IviSwtch Get Next Coercion Record

Each VI supported by the IviSwtch class driver has a corresponding property that determines the status code to return when status code simulation is enabled. The following table lists the status code properties and the VI for which they return a value. These properties are all of type `ViInt32`, non-readable, and non-channel based.



Note In the following table, the literal string `NISSWTCH_ATTR_` precedes all property names.

IviSwtch Properties for Status Simulation

Property	VI
INIT_STATUS	IviSwtch Initialize , IviSwtch Initialize With Options
CLOSE_STATUS	IviSwtch Close
RESET_STATUS	IviSwtch Reset
SELF_TEST_STATUS	IviSwtch Self-Test
ERROR_QUERY_STATUS	IviSwtch Error-Query
ERROR_MESSAGE_STATUS	IviSwtch Error Message
REVISION_QUERY_STATUS	IviSwtch Revision Query
RESET_DEFAULT_STATUS	IviSwtch Reset With Defaults
DISABLE_STATUS	IviSwtch Disable
INVALIDATE_STATUS	IviSwtch Invalidate All Attributes
GET_CH_NAME_STATUS	IviSwtch Get Channel Name
CAN_CONNECT_STATUS	IviSwtch Can Connect Channels
CONNECT_STATUS	IviSwtch Connect Channels
DISCONNECT_STATUS	IviSwtch Disconnect Channels
DISCONNECT_ALL_STATUS	IviSwtch Disconnect All Channels
GET_PATH_STATUS	IviSwtch Get Path
IS_DEBOUNCED_STATUS	IviSwtch Switch Is Debounced
SET_PATH_STATUS	IviSwtch Set Path
WAIT_FOR_DEBOUNCE_STATUS	IviSwtch Wait For Debounce
ABORT_SCAN_STATUS	IviSwtch Abort Scan
CONFIGURE_SCAN_LIST_STATUS	IviSwtch Configure Scan List
CONFIGURE_SCAN_TRIGGER_STATUS	IviSwtch Configure Scan Trigger
INITIATE_SCAN_STATUS	IviSwtch Initiate Scan

IS_SCANNING_STATUS	IviSwtch Switch Is Scanning
SET_CONTINUOUS_SCAN_STATUS	IviSwtch Set Continuous Scan
WAIT_FOR_SCAN_COMPLETE_STATUS	IviSwtch Wait For Scan To Complete
SEND_SOFTWARE_TRIGGER_STATUS	IviSwtch Send Software Trigger

Important Information

[Warranty](#)

[Copyright](#)

[Trademarks](#)

[Patents](#)

[Warning Regarding Use of NI Products](#)

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action

accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the patents.txt file on your CD, or ni.com/patents.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about [National Instruments trademarks](#).

FireWire® is the registered trademark of Apple Computer, Inc.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix® and Tek are registered trademarks of Tektronix, Inc.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR

APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- [Support](#)—Online technical support resources at ni.com/support include the following:
 - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable [KnowledgeBase](#), [product manuals](#), step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
 - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the [NI Discussion Forums](#) at forums.ni.com. National Instruments Applications Engineers make sure every question receives an answer. For information about other [technical support options](#) in your area, visit ni.com/services or [contact](#) your local office at ni.com/contact.
- [Training and Certification](#)—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- [System Integration](#)—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your [local office](#) or NI corporate headquarters. You also can visit the [Worldwide Offices](#) section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

error in

- error in (no error)** The error in cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- status** The status Boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- code** The code input identifies the error or warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- source** The source string describes the origin of the error or warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

[Error Codes](#)

error out

- error out** The error out cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- status** The status Boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- code** The code input identifies the error or warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

- source** The source string describes the origin of the error or warning.

The pop-up option Explain Error (or Explain Warning) gives more information about the error displayed.

[Error Codes](#)

ClassPrefix Channel Count

Inherent Property

Data Type	Access	Applies to	Coercion	High Level Functions
<input type="checkbox"/>	RO	N/A	None	None

Description

Returns the number of available channels.

Property Node Path

ClassPrefix»Inherent IVI Settings»User Options»Channel Count

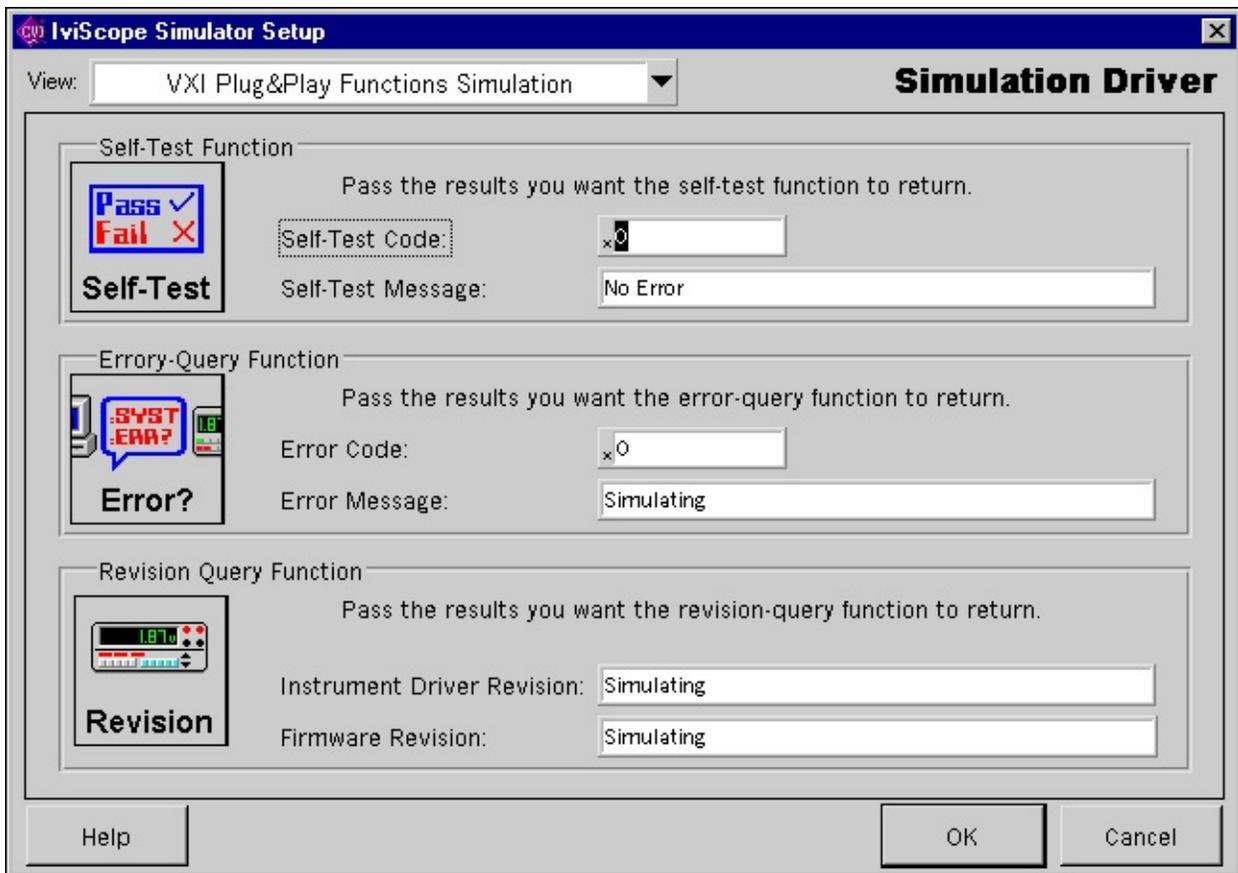
Interactive Simulation

In all class drivers, the *VXIplug&play*-compliant VIs and functions have output parameters that return values or strings from the instrument.

VXIplug&play-compliant VIs	
<ul style="list-style-type: none"> • CLASSPREFIX_self-test 	
<ul style="list-style-type: none"> • CLASSPREFIX_revision_query 	<ul style="list-style-type: none"> • CLASSPREFIX_error_query

During simulation, the simulation driver can create simulated data for the output parameters of these functions. This panel allows you to configure the simulation data to return from these functions.

You can access this view in the IviDmm, the IviDCPwr, or the IviScope Simulator Setup dialog boxes. In this case, you see the Simulator Setup dialog box for IviScope. Scroll down to see a description of each control.



IviScope Simulator Setup Dialog Box

The Simulator Setup dialog boxes for the different class drivers have common features. Each Simulator Setup dialog box has a View ring control in the upper left-hand corner. Use the View ring control to select a feature of the simulation driver to configure. All the dialog boxes have views to configure the simulation of the *VXIplug&play* functions and the simulation of function status codes. For classes that take measurements, such as IviDmm, IviScope, and IviDCPwr, the Simulator Setup dialog box also contains a Measurement Data Simulation view.

Because you can fully configure the simulation driver in the Simulator Setup dialog box at run time, you do not have to use MAX to specify attribute values in the Initial Settings tab of the Simulation Driver Sessions dialog box. If you do specify the initial settings, however, the values that you specify appear as the initial values in the Simulator Setup dialog box each time you call *ClassPrefix_init*.

Self-Test

Use the Result Code and Message controls to specify the values you want the driver's ClassPrefix Self-Test VI or *ClassPrefix_self_test* function to return in its output parameters.

Error-Query

Use the Error Code and Message controls to specify the values you want the driver's ClassPrefix Error-Query VI or *ClassPrefix_error_query* function to return in its output parameters.

Revision Query

Use the Instrument Driver Revision and Firmware Revision controls to specify the values you want the driver's ClassPrefix Revision Query VI or *ClassPrefix_revision_query* function to return in its output parameters.

Status Code Simulation

You use the dialog box shown below to configure the status code for a simulation. You can access this view in the IviDmm, the IviDCPwr, or the IviScope Simulator Setup dialog boxes. In this case, you see the Simulator Setup dialog box for IviScope.

To configure the status code simulation, select **Status Code Simulation** from the View ring control.

Scroll down to see a description of each control.



Status Code Simulation View

The Status Code Simulation view is the same for all simulation drivers.

The Status Code Simulation view has the following controls:

- **Simulate Status Codes**

This control lists all of the class driver functions that the selected instrument specific driver driver supports. The list also indicates each function's associated simulation status code in both macro name form and in hexadecimal value form. You use the **Status Code Macro** and **Custom Status Code** controls to modify simulation status codes.

Setting this control has the same effect as configuring the *PREFIX_ATTR_SIMULATE_STATUS_CODES* attribute on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»nisIviClass** in MAX.

- **Status Code Macro**

This control allows you to change the simulation status code of the currently selected function. Setting this control to the custom status code option sets the value you placed in the **Custom Status Code** control as the function's simulated status code.

If you select Custom Status Code, you can configure a custom status code in the Custom Status Code control.

- **Custom Status Code**

Lets you specify a custom status code to simulate for a particular function.

- **Reset all to VI_SUCCESS**

This control resets all of the class driver functions' simulation status codes to VI_SUCCESS (0).

Configuring simulated status codes with this dialog box has the same effect as configuring the corresponding simulated status code attributes on the Initial Settings tab of **IVI Drivers»Advanced»Simulation Driver Sessions»*nisIviClass*** in MAX.

Related Topics

[Properties for Controlling IviDCPwr Simulation](#)

[Properties for Controlling IviDmm Simulation](#)

[Properties for Controlling IviFgen Simulation](#)

[Properties for Controlling IviScope Simulation](#)

[Properties for Controlling IviSwtch Simulation](#)

Diagram of Standard Waveforms



Diagram of Standard Waveforms

VI_TRUE

Description

True

Defined Value

1

VI_FALSE

Description

False

Defined Value

0

IVISCOPE_VAL_AC_LINE_EITHER

Configures the oscilloscope to trigger on either positive or negative slope zero crossings of the network supply voltage.

Defined Value: 3

IVISCOPE_VAL_AC_LINE_POSITIVE

Configures the oscilloscope to trigger on positive slope zero crossings of the network supply voltage.

Defined Value: 1

IVISCOPE_VAL_AC_LINE_NEGATIVE

Configures the oscilloscope to trigger on negative slope zero crossings of the network supply voltage.

Defined Value: 2

Time Domain Measurements

In certain situations the spectrum analyzer may be used to take time domain rather than frequency domain measurements. Typical time domain measurements are oscilloscope traces, but when a spectrum analyzer is taking a time domain measurement, the amplitude of only one frequency is measured. To set the spectrum analyzer to a time domain mode, the start and stop frequency have to be set to the same frequency; the frequency of interest. Sweeping with the start and stop set to the same frequency causes the analyzer to take many independent measurements of that frequency over time (in the time domain). This is very useful in determining such things as the change in output power of a transmitter as it warms up. Another use is to capture a peak of a pulsed signal at a particular frequency.

IviSwtch Trigger Input Values

IVISWTCH_VAL_IMMEDIATE Immediate (Default)

IVISWTCH_VAL_EXTERNAL External

IVISWTCH_VAL_SOFTWARE_TRIG Software Trigger Function

IVISWTCH_VAL_TTL0PXI TRIG0 or VXI TTL0

IVISWTCH_VAL_TTL1PXI TRIG1 or VXI TTL1

IVISWTCH_VAL_TTL2PXI TRIG2 or VXI TTL2

IVISWTCH_VAL_TTL3PXI TRIG3 or VXI TTL3

IVISWTCH_VAL_TTL4PXI TRIG4 or VXI TTL4

IVISWTCH_VAL_TTL5PXI TRIG5 or VXI TTL5

IVISWTCH_VAL_TTL6PXI TRIG6 or VXI TTL6

IVISWTCH_VAL_TTL7PXI TRIG7 or VXI TTL7

IVISWTCH_VAL_ECL0VXI ECL0

IVISWTCH_VAL_ECL1VXI ECL1

IVISWTCH_VAL_PXI_STAR PXI Star

IVISWTCH_VAL_RTSL_0 RTSL line 0

IVISWTCH_VAL_RTSL_1 RTSL line 1

IVISWTCH_VAL_RTSL_2 RTSL line 2

IVISWTCH_VAL_RTSL_3 RTSL line 3

IVISWTCH_VAL_RTSL_4 RTSL line 4

IVISWTCH_VAL_RTSL_5 RTSL line 5

IVISWTCH_VAL_RTSL_6 RTSL line 6



Note For all of the following triggers, the switch module waits until it receives a trigger on the specified trigger line before processing the next entry in the scan list. . .

- (1) IVISWTCH_VAL_IMMEDIATE Immediate Trigger.
- (2) IVISWTCH_VAL_EXTERNAL External Trigger.
- (3) IVISWTCH_VAL_SOFTWARE_TRIG.
- (111) IVISWTCH_VAL_TTL.

- (112) IVISWTCH_VAL_TTL.
- (113) IVISWTCH_VAL_TTL.
- (114) IVISWTCH_VAL_TTL.
- (115) IVISWTCH_VAL_TTL.
- (116) IVISWTCH_VAL_TTL.
- (117) IVISWTCH_VAL_TTL.
- (118) IVISWTCH_VAL_TTL.
- (119) IVISWTCH_VAL_ECL.
- (120) IVISWTCH_VAL_ECL.
- (125) IVISWTCH_VAL_PXI_STA.
- (140) IVISWTCH_VAL_RTSI_0.
- (141) IVISWTCH_VAL_RTSI_1.
- (142) IVISWTCH_VAL_RTSI_2.
- (143) IVISWTCH_VAL_RTSI_.
- (144) IVISWTCH_VAL_RTSI_4.
- (145) IVISWTCH_VAL_RTSI_.
- (146) IVISWTCH_VAL_RTSI_6

IviSwtch Scan Adv Output Values

IVISWTCH_VAL_NONENTrigger (Default)

IVISWTCH_VAL_EXTERNALExternal

IVISWTCH_VAL_GPIB_SRQGPIB Service Request

IVISWTCH_VAL_TTL0PXI TRIG0 or VXI TTL0

IVISWTCH_VAL_TTL1PXI TRIG1 or VXI TTL1

IVISWTCH_VAL_TTL2PXI TRIG2 or VXI TTL2

IVISWTCH_VAL_TTL3PXI TRIG3 or VXI TTL3

IVISWTCH_VAL_TTL4PXI TRIG4 or VXI TTL4

IVISWTCH_VAL_TTL5PXI TRIG5 or VXI TTL5

IVISWTCH_VAL_TTL6PXI TRIG6 or VXI TTL6

IVISWTCH_VAL_TTL7PXI TRIG7 or VXI TTL7

IVISWTCH_VAL_ECL0VXI ECL0

IVISWTCH_VAL_ECL1VXI ECL1

IVISWTCH_VAL_PXI_STARPXI Star

IVISWTCH_VAL_RTSL_0RTSL line 0

IVISWTCH_VAL_RTSL_1RTSL line 1

IVISWTCH_VAL_RTSL_2RTSL line 2

IVISWTCH_VAL_RTSL_3RTSL line 3

IVISWTCH_VAL_RTSL_4RTSL line 4

IVISWTCH_VAL_RTSL_5RTSL line 5

IVISWTCH_VAL_RTSL_6RTSL line 6



Note (0) IVISWTCH_VAL_NONEThe switch module does not produce a Scan Advanced Output trigger.



Note For all of the following triggers, the switch module waits until it receives a trigger on the specified trigger line before processing the next entry in the scan list. . .

- (1) IVISWTCH_VAL_IMMEDIATE Immediate Trigger.
- (2) IVISWTCH_VAL_EXTERNAL External Trigger.

- (3) IVISWTCH_VAL_SOFTWARE_TRIG.
- (111) IVISWTCH_VAL_TTL.
- (112) IVISWTCH_VAL_TTL.
- (113) IVISWTCH_VAL_TTL.
- (114) IVISWTCH_VAL_TTL.
- (115) IVISWTCH_VAL_TTL.
- (116) IVISWTCH_VAL_TTL.
- (117) IVISWTCH_VAL_TTL.
- (118) IVISWTCH_VAL_TTL.
- (119) IVISWTCH_VAL_ECL.
- (120) IVISWTCH_VAL_ECL.
- (125) IVISWTCH_VAL_PXI_STA.
- (140) IVISWTCH_VAL_RTSL_0.
- (141) IVISWTCH_VAL_RTSL_1.
- (142) IVISWTCH_VAL_RTSL_2.
- (143) IVISWTCH_VAL_RTSL_3.
- (144) IVISWTCH_VAL_RTSL_4.
- (145) IVISWTCH_VAL_RTSL_.
- (146) IVISWTCH_VAL_RTSL_6

Branch Offices

Office	Telephone Number
Australia	02 612 9672 8846
Austria	43 0 662 45 79 90 0
Belgium	32 0 2 757 00 20
Brazil	55 11 3262 3599
Canada (Calgary)	403 274 9391
Canada (Montreal)	514 288 5722
Canada (Ottawa)	613 233 5949
Canada (Québec)	514 694 8521
Canada (Toronto)	905 785 0085
Canada (Vancouver)	514 685 7530
China	86 21 6555 7838
Czech Republic	420 2 2423 5774
Denmark	45 45 76 26 00
Finland	385 0 9 725 725 11
France	33 0 1 48 14 24 24
Germany	49 0 89 741 31 30
Greece	30 2 10 42 96 427
India	91 80 51190000
Israel	972 0 3 6393737
Italy	39 02 413091
Japan	81 3 5472 2970
Korea	82 02 3451 3400
Malaysia	603 9131 0918
Mexico	001 800 010 0793
Netherlands	31 0 348 433 466
New Zealand	64 09 914 0488
Norway	47 0 32 27 73 00

Poland	48 0 22 3390 150
Portugal	351 210 311 210
Russia	7 095 238 7139
Singapore	65 6226 5886
Slovenia	386 3 425 4200
South Africa	27 0 11 805 8197
Spain	34 91 640 0085
Sweden	46 0 8 587 895 00
Switzerland	41 56 200 51 51
Taiwan	886 2 2528 7227
Thailand	662 992 7519
United Kingdom	44 0 1635 523545
United States (Corporate)	512 683 0100