DAQmx - Data Acquisition VIs and Functions

January 2008, 370469L-01

Use the DAQmx - Data Acquisition VIs with NI-DAQ and NI-SWITCH hardware devices to develop instrumentation, acquisition, and control applications. Refer to the *NI-DAQ Readme* for a complete listing of devices that NI-DAQmx supports.

Palette Object	Description
DAQ Assistant Express VI	Creates, edits, and runs tasks using NI-DAQmx.
<u>DAQmx</u> <u>Channel</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all devices installed in the system.
<u>DAQmx</u> <u>Clear</u> Task	Clears the <u>task</u> . Before clearing, this VI stops the task, if necessary, and releases any resources the task reserved. You cannot use a task after you clear it unless you recreate the task.
<u>DAQmx</u> <u>Create</u> <u>Virtua</u> l <u>Channel</u>	Creates a <u>virtual channel</u> or set of virtual channels and adds them to a <u>task</u> . The instances of this <u>polymorphic VI</u> correspond to the I/O type of the channel, such as analog input, digital output, or counter output; the measurement or generation to perform, such as temperature measurement, voltage generation, or event counting; and in some cases, the sensor to use, such as a thermocouple or RTD for temperature measurements.
DAQmx Global Channel Constant	Lists all <u>virtual channels</u> you create and save using the <u>DAQ</u> <u>Assistant</u> . Select Browse to select multiple channels. Right- click the constant and select I/O Name Filtering from the shortcut menu to <u>limit the channels</u> that the constant displays and to limit what you can enter in the constant.
<u>DAQmx</u>	A Property Node with the <u>DAQmx Read</u> class preselected.

<u>Read</u> Property Node	Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Read</u>	Reads samples from the <u>task</u> or <u>virtual channels</u> you specify. The instances of this <u>polymorphic VI</u> specify what format of samples to return, whether to read a single sample or multiple samples at once, and whether to read from one or multiple channels.
<u>DAQmx</u> <u>Start</u> <u>Task</u>	Transitions the <u>task</u> to the running <u>state</u> to begin the measurement or generation. <u>Using this VI</u> is required for some applications and is optional for others.
<u>DAQmx</u> <u>Stop</u> Task	Stops the <u>task</u> and returns it to the <u>state</u> the task was in before the <u>DAQmx Start Task</u> VI ran or the <u>DAQmx Write</u> VI ran with the autostart input set to TRUE.
<u>DAQmx</u> <u>Task</u> <u>Name</u> Constant	Lists all <u>tasks</u> you create and save by using the <u>DAQ</u> <u>Assistant</u> . You cannot use this constant to select multiple tasks. Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the tasks</u> that the constant displays and to limit what you can enter in the constant.
DAQmx Timing Property Node	A Property Node with the <u>DAQmx Timing</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> Timing	Configures the number of samples to acquire or generate and creates a buffer when needed. The instances of this polymorphic VI correspond to the type of timing to use for the task.
DAQmx Trigger Property Node	A Property Node with the <u>DAQmx Trigger</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> Trigger	Configures <u>triggering</u> for the task. The instances of this <u>polymorphic VI</u> correspond to the trigger and trigger type to

	configure.
<u>DAQmx</u> <u>Wait</u> <u>Until</u> Done	Waits for the measurement or generation to complete. Use this VI to <u>ensure that the specified operation is complete</u> before you stop the task.
<u>DAQmx</u> <u>Write</u> Property <u>Node</u>	A Property Node with the <u>DAQmx Write</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> Write	Writes samples to the <u>task</u> or <u>virtual channels</u> you specify. The instances of this <u>polymorphic VI</u> specify the format of the samples to write, whether to write one or multiple samples, and whether to write to one or multiple channels.

Subpalette	Description
DAQmx Advanced VIs and Functions	Use the DAQmx Advanced VIs and functions to access advanced and miscellaneous features of NI-DAQmx.
DAQmx Advanced Task Options VIs and Functions	Use the DAQmx Advanced Task Options VIs and functions for advanced configuration and control of <u>tasks</u> .
DAQmx Device Configuration VIs and Functions	Use the DAQmx Device Configuration VIs and functions for hardware-specific configuration and control.
DAQmx Real Time VIs and Function	Use the DAQmx Real Time VIs and function to configure and perform real-time operations.

To comment on National Instruments documentation, refer to the <u>National</u> <u>Instruments Web site</u>.

© 2003–2008 National Instruments Corporation. All Rights Reserved.

Converting a DAQ Assistant Express VI to an NI-DAQmx Task

When you use the <u>DAQ Assistant Express VI</u> to configure a task, the task is local to the application and cannot be used in other applications. You can convert a DAQ Assistant Express VI to an NI-DAQmx task saved in MAX, which makes the task globally accessible to all applications and allows you to use a <u>DAQmx task name</u> constant or control to edit the task and generate code for the task.



Note If you are working within a project in LabVIEW 8.0, converting the DAQ Assistant Express VI to a task saves the task to the project.

Complete the following steps to convert the DAQ Assistant Express VI to an NI-DAQmx task.

- 1. Right-click the DAQ Assistant Express VI and select **Convert to NI-DAQmx Task** from the shortcut menu.
- 2. The DAQ Assistant launches, and you can modify the task, if necessary.
- 3. Click the **OK** button.
- 4. NI-DAQmx saves the task to MAX or the LabVIEW project and replaces the DAQ Assistant Express VI with a DAQmx task name constant.

NI-DAQmx Related Documentation

The following documents contain information that you might find helpful as you use this help file. You must install the PDFs to access them from this help file. You must have Adobe Reader with Search and Accessibility 5.0.5 or later installed to view the PDFs. Refer to the <u>Adobe Systems</u> <u>Incorporated Web site</u> to download Adobe Reader. Refer to the <u>National</u> <u>Instruments Product Manuals Library</u> for updated documentation resources.

• <u>NI-DAQmx Help</u>—Use this help file to learn about common NI-DAQmx applications, measurement fundamentals, NI-DAQmx key concepts, and device-specific programming considerations.

NI-DAQmx Software Events

NI-DAQmx Software events provide an asynchronous notification mechanism for a set of <u>NI-DAQmx hardware signals and conditions</u>. Using event-based programming, you can write an application that continues to perform work while waiting for data without resorting to developing a multithreaded application.

To use NI-DAQmx software events in LabVIEW, wire a task or global channel to the **event source** input of the <u>Register for Events</u> node when you <u>dynamically register</u> an event.



Note NI-DAQmx software events are not supported for NI-DAQmx simulated devices.

Properties Settable at Task Run Time

You can set some properties while a task is running. These properties vary by device and/or device family.

- AO Series Devices
- DSA Devices
- E Series, M Series, and S Series Devices
- <u>Static DIO Devices</u>
- <u>TIO Devices</u>

E Series Devices

You can set the following properties on E Series, M Series, and S Series devices while the task is running. If a property is not listed here, you cannot set it while the task is running.

DAQmx Channel

Analog Output:General Properties:DAC:Reference Voltage:Connect DAC Reference to Ground(E Series and M Series only) Digital Output:Tristate Counter Output:Pulse:Time:High Time Counter Output:Pulse:Time:Low Time Counter Output:Pulse:Frequency:Frequency Counter Output:Pulse:Frequency:Duty Cycle Counter Output:Pulse:Ticks:High Ticks Counter Output:Pulse:Ticks:Low Ticks

DAQmx Timing

Sample Clock:Rate (Analog Output only)

AO Series Devices

You can set the following properties on AO Series devices while the task is running. If a property is not listed here, you cannot set it while the task is running.

DAQmx Channel

Analog Output:General Properties:DAC:Reference Voltage:Connect DAC Reference to Ground Digital Output:Tristate Counter Output:Pulse:Time:High Time Counter Output:Pulse:Time:Low Time Counter Output:Pulse:Frequency:Frequency Counter Output:Pulse:Frequency:Duty Cycle Counter Output:Pulse:Ticks:High Ticks Counter Output:Pulse:Ticks:Low Ticks

DAQmx Timing

Sample Clock:Rate (Analog Output only)

DSA Devices

You cannot set any properties on DSA devices while the task is running.

Static DIO Devices

You cannot set any properties on static DIO devices while the task is running.

Using Traditional NI-DAQ (Legacy) and NI-DAQmx in the Same Application

You can use both Traditional NI-DAQ (Legacy) and NI-DAQmx in the same computer, but there are some restrictions. After using a device in NI-DAQmx, you must unreserve all NI-DAQmx tasks that use that device before you can use that device through Traditional NI-DAQ (Legacy). After using a device in Traditional NI-DAQ (Legacy), you must reset the device before you can use that device in NI-DAQmx. Refer to the *DAQ Getting Started Guide* for additional information.

Refer to <u>ni.com/support</u> for details and instructions about the following topics:

- How to add NI-DAQmx code to a Traditional NI-DAQ (Legacy) application
- How to run both Traditional NI-DAQ (Legacy) applications and NI-DAQmx applications that use the same device

DAQmx Clear Task

Clears the <u>task</u>. Before clearing, this VI stops the task, if necessary, and releases any resources the task reserved. You cannot use a task after you clear it unless you recreate the task.

If you use the <u>DAQmx Create Task</u> VI or the <u>DAQmx Create Virtual</u> <u>Channel</u> VI within a loop, use this VI within the loop after you are finished with the task to avoid allocating unnecessary memory. Refer to <u>Task</u> <u>Creation and Destruction</u> for more information about when to use this VI.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task in** is the name of the task to clear. Unlike the **task/channels in** input on some VIs, you cannot wire a virtual channel to this input.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Create Virtual Channel

Creates a <u>virtual channel</u> or set of virtual channels and adds them to a <u>task</u>. The instances of this <u>polymorphic VI</u> correspond to the I/O type of the channel, such as analog input, digital output, or counter output; the measurement or generation to perform, such as temperature measurement, voltage generation, or event counting; and in some cases, the sensor to use, such as a thermocouple or RTD for temperature measurements.

If you use this VI within a loop without specifying a **task in**, NI-DAQmx creates a new task in each iteration of the loop. Use the <u>DAQmx Clear</u> <u>Task</u> VI within the loop after you are finished with the task to avoid allocating unnecessary memory. Refer to <u>Task Creation and Destruction</u> for more information about when NI-DAQmx creates tasks and when LabVIEW automatically destroys tasks.

The <u>DAQmx Channel</u> properties include additional channel configuration options.

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

Select an instance	•	1
--------------------	---	---

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

AI Voltage

Creates channel(s) to <u>measure voltage</u>. If the measurement requires the use of internal excitation or you need excitation to scale the voltage, use the <u>AI Custom Voltage with Excitation</u> instance of this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return voltage measurements.

Custom Scale	value, you must wire a custom scale name to the custom scale name input.
(10065)	
V (10348)	Volts.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> terminal configuration for the channel.
Differential (10106)	<u>Differential mode</u> .
NRSE (10078)	Non-referenced single-ended mode.

Pseudodifferential (12529)	<u>Pseudodifferential mode</u> .
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Voltage RMS

Creates channel(s) to measure voltage RMS, the average (mean) power of the acquired voltage.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return voltage measurements.

FromUnits a custom scale specifies. If you select thisCustomvalue, you must wire a custom scale name to the

Scale (10065)	custom scale name input.
V (10348)	Volts.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.
Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential	Pseudodifferential mode.

(12529)	
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Temp TC

Creates channel(s) that use a thermocouple to measure temperature.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **thermocouple type** specifies the type of thermocouple connected

to the channel. <u>Thermocouple types</u> differ in composition and measurement range.

B (10047)	B-type thermocouple.
E (10055)	E-type thermocouple.
J (10072)	J-type thermocouple.
K (10073)	K-type thermocouple.
N (10077)	N-type thermocouple.
R (10082)	R-type thermocouple.
S (10085)	S-type thermocouple.
T (10086)	T-type thermocouple.

- **cjc value** specifies in **units** the temperature of the <u>cold junction</u> if you set **cjc source** to **Constant Value**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **cjc source** specifies the source of <u>cold-junction compensation</u>.

Channel (10113)	You must use the cjc channel input to specify a channel.
Constant Value (10116)	You must use the cjc value input to specify the cold- junction temperature.
Internal (10200)	Use a cold-junction compensation channel built into the terminal block. If no such channel is available, this VI returns an error.

cjc channel specifies the channel that acquires the temperature of the thermocouple <u>cold-junction</u> if you set **cjc source** to **Channel**. You can use a global channel or another virtual channel already in the task. If the channel is a temperature channel, NI-DAQmx acquires the temperature in the correct units. Other channel types, such as a resistance channel with a custom sensor, must use a custom scale to scale values to degrees Celsius.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Temp RTD

Г

Creates channel(s) that use an RTD to measure temperature.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **resistance configuration** specifies the number of wires to use for

resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

current excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

rtd type specifies the <u>type of RTD</u> connected to the channel.

Custom (10137)	You must use the <u>Analog</u> <u>Input:Temperature:RTD:Custom:A</u> , <u>Analog</u> <u>Input:Temperature:RTD:Custom:B</u> , and <u>Analog</u> <u>Input:Temperature:RTD:Custom:C</u> properties to supply the coefficients for the Callendar-Van Dusen equation.
Pt3750 (12481)	Pt3750.
Pt3851 (10071)	Pt3851.
Pt3911 (12482)	Pt3911.
Pt3916 (10069)	Pt3916.
Pt3920 (10053)	Pt3920.
Pt3928	Pt3928.

(12483)

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **r0** is the sensor resistance in ohms at 0 degrees Celsius. The <u>Callendar-Van Dusen equation</u> requires this value. Refer to the sensor documentation to determine this value.
- current excitation value specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Thermistor lex

Creates channel(s) that use a <u>thermistor</u> to <u>measure temperature</u>. Use this instance when the thermistor requires current excitation.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- resistance configuration specifies the number of wires to use for

resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

- thermistor characteristics contains the constants for the Steinhart-Hart thermistor equation. Refer to the sensor documentation to determine values for these constants.
 - **A** is the A constant for the equation.
 - **B** is the B constant for the equation.
 - **C** is the C constant for the equation.

current excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the
shortcut menu for more information about the error.

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

AI Thermistor Vex

Creates channel(s) that use a <u>thermistor</u> to <u>measure temperature</u>. Use this instance when the thermistor requires voltage excitation.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.

resistance configuration specifies the number of wires to use for resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

- thermistor characteristics contains the constants for the Steinhart-Hart thermistor equation. Refer to the sensor documentation to determine values for these constants.
 - **A** is the A constant for the equation.
 - **B** is the B constant for the equation.
 - **C** is the C constant for the equation.

voltage excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **r1** specifies in ohms the value of the reference resistor.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- voltage excitation value specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Al Current

Creates channel(s) to measure current.



- I/0 | task in specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- I/0 | **physical channels** specifies the names of the physical channels to use to create virtual channels. The DAQmx physical channel constant lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a list or range of physical channels to this input. If you have an array of physical channels, use the DAOmx Flatten Channel String VI to convert the array to a list.
- abc **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAOmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx automatically assigns names to the virtual channels.

132 units specifies the units to use to return current measurements.

Amps

(10342)	
From	Units a custom scale specifies. If you select this
Custom	value, you must wire a custom scale name to the
Scale	custom scale name input.
(10065)	

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> terminal configuration for the channel.
Differential (10106)	<u>Differential mode</u> .

NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **shunt resistor location** specifies the location of the shunt resistor. For devices with built-in shunt resistors, specify the location as **Internal**. For devices that do not have built-in shunt resistors, you must attach an external one, set this input to **External** and use the **external shunt resistor value** input to specify the value of the resistor.

default (-1)	NI-DAQmx selects Internal for devices with an internal shunt resistor. NI-DAQmx selects External for devices without an internal shunt resistor.
External (10167)	Use a shunt resistor external to the device. You must use the external shunt resistor value input to specify the value of the shunt resistor.
Internal (10200)	Use the built-in shunt resistor of the device.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **external shunt resistor value** specifies in ohms the resistance of an external shunt resistor.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

AI Current RMS

Creates a channel to measure current RMS, the average (mean) power of the acquired current.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return current measurements.

Amps (10342)	Amperes.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.
Differential	Differential mode.

(10106)	
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **shunt resistor location** specifies the location of the shunt resistor. For devices with built-in shunt resistors, specify the location as **Internal**. For devices that do not have built-in shunt resistors, you must attach an external one, set this input to **External** and use the **external shunt resistor value** input to specify the value of the resistor.

default (-1)	NI-DAQmx selects Internal for devices with an internal shunt resistor. NI-DAQmx selects External for devices without an internal shunt resistor.
External (10167)	Use a shunt resistor external to the device. You must use the external shunt resistor value input to specify the value of the shunt resistor.
Internal (10200)	Use the built-in shunt resistor of the device.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **external shunt resistor value** specifies in ohms the resistance of an external shunt resistor.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

AI Resistance

Creates channel(s) to measure resistance.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return resistance measurements.

From Units a custom scale specifies. If you select this

Custom Scale (10065)	value, you must wire a custom scale name to the custom scale name input.
Ohms (10384)	Ohms.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- resistance configuration specifies the number of wires to use for resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

I/0

custom scale name specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.

current excitation source specifies the source of excitation.

Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Supply no excitation to the channel.
-

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

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

Al Strain Gage

Creates channel(s) to measure strain.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return strain measurements.

FromUnits a custom scale specifies. If you select thisCustomvalue, you must wire a custom scale name to the

Scale (10065)	custom scale name input.
Strain (10299)	Strain.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **bridge information** specifies information about the bridge configuration and measurement.
 - **strain configuration** specifies the <u>bridge configuration</u> for the strain gages.

Full Bridge I (10183)	Full Bridge I.
Full Bridge II (10184)	Full Bridge II.
Full Bridge III (10185)	Full Bridge III.
Half Bridge I (10188)	Half Bridge I.
Half Bridge II (10189)	Half Bridge II.
Quarter Bridge I (10271)	Quarter Bridge I.
Quarter Bridge II (10272)	Quarter Bridge II.

voltage excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **lead wire resistance** is the amount of resistance in ohms in the lead wires. Ideally, this value is the same for all leads.
- **initial bridge voltage** specifies in volts the output voltage of the bridge in the unloaded condition. NI-DAQmx subtracts this value from any measurements before applying scaling equations. Perform a voltage measurement on the bridge with no strain applied to determine this value.
- **strain gage information** contains information about the strain gage and measurement.

DBL

gage factor specifies the sensitivity of the strain gages and relates the change in electrical resistance to the change in strain. Each gage in the bridge must have the same gage factor. Refer to the sensor documentation to determine this value.

- **nominal gage resistance** is the resistance in ohms of the gages in an unstrained position. Each gage in the bridge must have the same nominal gage resistance. The resistance across arms of the bridge that do not have strain gages must also be the same as the nominal gage resistance. Refer to the sensor documentation to determine this value.
- **poisson ratio** is the ratio of lateral strain to axial strain in the material you are measuring.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Freq Voltage

Creates channel(s) that use a frequency-to-voltage converter to <u>measure</u> <u>frequency</u>.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return frequency measurements.

From Units a custom scale specifies. If you select this

Custom Scale (10065)	value, you must wire a custom scale name to the custom scale name input.
Hz (10373)	Hertz.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.

- **threshold level** specifies in volts the level at which to recognize waveform repetitions. You should select a voltage level that occurs only once within the entire period of a waveform. You also can select a voltage that occurs only once while the voltage rises or falls.
- **hysteresis** specifies in volts a window below **level**. The input voltage must pass below **threshold level** minus **hysteresis** before NI-DAQmx recognizes a waveform repetition. Hysteresis can improve measurement accuracy when the signal contains noise or jitter.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI LVDT

Creates channel(s) that use an LVDT to measure linear position.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return linear position measurements from the channel.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Inches (10379)	Inches.
Meters (10219)	Meters.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **voltage excitation wire mode** is the number of leads on the sensor. Some sensors require you to tie leads together to create a four- or five- wire sensor. Refer to the sensor documentation for

more information.

4-Wire (4)	Four wires.
5-Wire (5)	Five wires.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **sensitivity** is the sensitivity of the sensor. This value is in the units you specify with the **sensitivity units** input. Refer to the sensor documentation to determine this value.
- **sensitivity units** specifies the units of the **sensitivity** input.

mVolts/Volt/0.001 Inch (12505)	mVolts/Volt/0.001 Inch.
mVolts/Volt/mMeter (12506)	mVolts/Volt/mMeter.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **voltage excitation frequency** specifies in hertz the excitation

frequency that the sensor requires. Refer to the sensor documentation to determine this value.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI RVDT

Creates channel(s) that use an <u>RVDT</u> to <u>measure angular position</u>.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return angular position measurements from the channel.

Degrees (10146)	Degrees.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Radians (10273)	Radians.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **voltage excitation wire mode** is the number of leads on the sensor. Some sensors require you to tie leads together to create a four- or five- wire sensor. Refer to the sensor documentation for

more information.

4-Wire (4)	Four wires.
5-Wire (5)	Five wires.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **sensitivity** is the sensitivity of the sensor. This value is in the units you specify with the **sensitivity units** input. Refer to the sensor documentation to determine this value.
- **sensitivity units** specifies the units of the **sensitivity** input.

mVolts/Volt/Degree (12507) mVolts/Volt/Degree. **mVolts/Volt/Radian** (12508) mVolts/Volt/Radian.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **voltage excitation frequency** specifies in hertz the excitation

frequency that the sensor requires. Refer to the sensor documentation to determine this value.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AI Accelerometer

Creates channel(s) that use an accelerometer to measure acceleration.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return acceleration measurements from the channel.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
g (10186)	1 g is approximately equal to 9.81 m/s/s.
m/s^2 (12470)	Meters per second per second.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

	terminal configuration for the channel.
Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **sensitivity** is the sensitivity of the sensor. This value is in the units you specify with the **sensitivity units** input. Refer to the sensor documentation to determine this value.
- **sensitivity units** specifies the units of the **sensitivity** input.

mVolts/g (12509)	mVolts/g.
Volts/g (12510)	Volts/g.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

current excitation value specifies in amperes the amount of

excitation that the sensor requires. Refer to the sensor documentation to determine this value.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
AI Microphone

Creates channel(s) that use a microphone to measure sound pressure.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return sound pressure measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Pascals (10081)	Pascals.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum sound pressure level** is the maximum instantaneous sound pressure level you expect to measure. This value is in decibels, referenced to 20 micropascals.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.
Differential (10106)	Differential mode.

NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **microphone sensitivity** is the sensitivity of the microphone. Specify this value in mV/Pa.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

(checkmark) to indicate a warning or that no error occurred.

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

AI Custom Voltage with Excitation

Creates channel(s) to measure voltage. Use this instance for custom sensors that require excitation. You can use the excitation to scale the measurement.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels. units specifies the units to use to return voltage measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
V (10348)	Volts.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.

Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

bridge configuration specifies what type of Wheatstone bridge the sensor is.

Full Bridge (10182)	Sensor is a full bridge. If you set use excitation for scaling to TRUE, NI-DAQmx divides the measurement by the excitation value. Many sensors scale data to native units using scaling of volts per excitation.
Half Bridge (10187)	Sensor is a half bridge. If you set use excitation for scaling to TRUE, NI-DAQmx divides the measurement by the excitation value. Many sensors scale data to native units using scaling of volts per excitation.
No Bridge (10228)	Sensor is not a Wheatstone bridge.

QuarterSensor is a quarter bridge. If you set use excitation forBridgescaling to TRUE, NI-DAQmx divides the measurement(10270)by the excitation value. Many sensors scale data to
native units using scaling of volts per excitation.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **use excitation for scaling** specifies if NI-DAQmx divides the measurement by the excitation. You should typically set **use excitation for scaling** to TRUE for ratiometric transducers. If you set **use excitation for scaling** to TRUE, set **maximum value** and **minimum value** to reflect the scaling.

For example, if you expect to acquire a voltage between -5 and 5, and you use an excitation of 0.10 volts to scale the measurement, set **minimum value** to -50 and set **maximum value** to 50. If you set **bridge configuration** to **No Bridge**, **use excitation for scaling** has no effect on the measurement.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the

error.

AI Temp Built-In Sensor

Creates channel(s) that use the built-in sensor of a terminal block or device to measure temperature. On SCXI modules, for example, the built-in sensor could be the CJC sensor.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.

deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

AO Voltage

Creates channel(s) to generate voltage.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to generate voltage.

From	Units a custom scale specifies. If you select this
Custom	value, you must wire a custom scale name to the
Scale	custom scale name input.

(10065)	
V (10348)	Volts.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to generate.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to generate.
- **output terminal configuration** specifies the <u>output terminal</u> <u>configuration</u> for the channel.

Default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.
Differential (10106)	<u>Differential mode</u> .
Pseudodifferential (12529)	Pseudodifferential mode.

I/0 |

custom scale name specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

AO Current

Creates channel(s) to generate current.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to generate current.

Amps (10342)	Amperes.
From	Units a custom scale specifies. If you select this

Customvalue, you must wire a custom scale name to theScalecustom scale name input.(10065)

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Function Generation

Creates a channel for continually generating a waveform on the selected physical channel.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

type specifies the kind of waveform to generate.

Sine (14751)	Sine wave.
Triangle (14752)	Triangle wave.

Square (14753)	Square wave.
Sawtooth (14754)	Sawtooth wave.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **frequency** is the frequency of the waveform to generate in hertz.
- **amplitude** is the zero-to-peak amplitude of the waveform to generate in volts. Zero and negative values are valid.
- **offset** is the voltage offset of the waveform to generate.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

Digital Input

Creates channel(s) to <u>measure digital signals</u>. You can group digital <u>lines</u> into one <u>digital channel</u> or separate them into multiple digital channels. If you specify one or more entire <u>ports</u> in the **lines** input by using port physical channel names, you cannot separate the ports into multiple channels. To separate ports into multiple channels, use this VI multiple times with a different port each time.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **Ines** specifies the names of the digital lines or ports to use to create virtual channels. The <u>DAQmx physical channel constant</u> lists all lines and ports for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input. If you have an array of lines or ports, use the <u>DAQmx</u> <u>Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

Line grouping specifies how to group digital lines into one or more virtual channels. If you specify one or more entire ports with the lines input, you must set this input to one channel for all lines.

one channel for all lines (1)	Combine all digital lines into a single virtual channel.
one channel for each line (0)	Create a separate virtual channel for each digital line.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

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

Digital Output

Creates channel(s) to <u>generate digital signals</u>. You can group digital <u>lines</u> into one <u>digital channel</u> or separate them into multiple digital channels. If you specify one or more entire <u>ports</u> in **lines** input by using port physical channel names, you cannot separate the ports into multiple channels. To separate ports into multiple channels, use this VI multiple times with a different port each time.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **Ines** specifies the names of the digital lines or ports to use to create virtual channels. The <u>DAQmx physical channel constant</u> lists all lines and ports for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input. If you have an array of lines or ports, use the <u>DAQmx</u> <u>Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

Line grouping specifies how to group digital lines into one or more virtual channels. If you specify one or more entire ports with the lines input, you must set this input to one channel for all lines.

one channel for all lines (1)	Combine all digital lines into a single virtual channel.
one channel for each line (0)	Create a separate virtual channel for each digital line.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

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

CI Freq

Creates a channel to <u>measure the frequency of a digital signal</u>. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signal to the <u>default input terminal</u> of the counter unless you select a different input terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return frequency measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Hz (10373)	Hertz.
Ticks (10304)	Timebase ticks.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the maximum value you expect to measure.
- **minimum value** specifies in **units** the minimum value you expect to measure.
- **measurement method** specifies the method to use to calculate the period or frequency of the signal.

Frequency with 2 Counters (10157)	measure during measurement time .
Large Range with 2 Counters (10205)	Use one counter to divide the frequency of the input signal by divisor to create a lower-frequency signal that the second counter can more easily measure.
Low Frequency with 1 Counter (10105)	Use one counter that uses a constant timebase to measure the input signal.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **starting edge** specifies between which edges to measure the frequency or period of the signal.

Falling (10171)	Measure between falling edges.
Rising (10280)	Measure between rising edges.

- measurement time is the length of time in seconds to measure the frequency or period of the signal if measurement method is High Frequency with 2 Counters. Leave this input unwired if measurement method is not High Frequency with 2 Counters. Measurement accuracy increases with increased measurement time and with increased signal frequency. If you measure a highfrequency signal for too long, however, the count register could roll over, which results in an incorrect measurement.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- divisor is the value by which to divide the input signal when

measurement method is Large Range with 2 Counters. Leave this input unwired if **measurement method** is not Large Range with 2 Counters. The larger the divisor, the more accurate the measurement. However, too large a value could cause the count register to roll over, which results in an incorrect measurement.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Period

Creates a channel to <u>measure the period of a digital signal</u>. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signal to the <u>default input terminal</u> of the counter unless you select a different input terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return time or period

measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Seconds (10364)	Seconds.
Ticks (10304)	Timebase ticks.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the maximum value you expect to measure.
- **minimum value** specifies in **units** the minimum value you expect to measure.
- **measurement method** specifies the method to use to calculate the period or frequency of the signal.

High Frequency with 2 Counters (10157)	Use two counters to count pulses of the signal to measure during measurement time .
Large Range with 2 Counters (10205)	Use one counter to divide the frequency of the input signal by divisor to create a lower-frequency signal that the second counter can more easily measure.
Low Frequency with 1 Counter (10105)	Use one counter that uses a constant timebase to measure the input signal.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **starting edge** specifies between which edges to measure the frequency or period of the signal.

Falling (10171)	Measure between falling edges.
Rising (10280)	Measure between rising edges.

- measurement time is the length of time in seconds to measure the frequency or period of the signal if measurement method is High Frequency with 2 Counters. Leave this input unwired if measurement method is not High Frequency with 2 Counters. Measurement accuracy increases with increased measurement time and with increased signal frequency. If you measure a highfrequency signal for too long, however, the count register could roll over, which results in an incorrect measurement.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.

- divisor is the value by which to divide the input signal when measurement method is Large Range with 2 Counters. Leave this input unwired if measurement method is not Large Range with 2 Counters. The larger the divisor, the more accurate the measurement. However, too large a value could cause the count register to roll over, which results in an incorrect measurement.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Cnt Edges

Creates a channel to <u>count the number of rising or falling edges of a</u> <u>digital signal</u>. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signal to the <u>default input terminal</u> of the counter unless you select a different input terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI
or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **initial count** is the value from which to start counting.
- **count direction** specifies whether to increment or decrement the counter on each edge.

Count Down (10124)	Decrement counter.
Count Up (10128)	Increment counter.
Externally Controlled (10326)	The state of a digital line controls the count direction. Each counter has a <u>default count</u> <u>direction terminal</u> .

edge specifies on which edges of the input signal to increment or decrement the count.

Falling (10171)Count falling edges.Rising (10280)Count rising edges.

task out is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Pulse Width

Creates a channel to <u>measure the width of a digital pulse</u>. **starting edge** determines whether to measure a high pulse or low pulse. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signal to the <u>default input terminal</u> of the counter unless you select a different input terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return time or period measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Seconds (10364)	Seconds.
Ticks (10304)	Timebase ticks.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the maximum value you expect to measure.
- **minimum value** specifies in **units** the minimum value you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From**

Custom Scale.

starting edge specifies on which edge to begin measuring pulse width.

Falling (10171)	Begin measuring on the falling edge. The channel measures the width of low pulses.
Rising (10280)	Begin measuring on the rising edge. The channel measures the width of high pulses.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Semi Period

Creates a channel to <u>measure the time between state transitions of a</u> <u>digital signal</u>. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signal to the <u>default input terminal</u> of the counter unless you select a different input terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return time or period measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Seconds (10364)	Seconds.
Ticks (10304)	Timebase ticks.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the maximum value you expect to measure.
- **minimum value** specifies in **units** the minimum value you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Angular Encoder

Creates a channel that uses an <u>angular encoder</u> to <u>measure angular</u> <u>position</u>. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signals to the <u>default input terminals</u> of the counter unless you select different input terminals.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return angular position

measurements from the channel.

Degrees (10146)	Degrees.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Radians (10273)	Radians.
Ticks (10304)	Timebase ticks.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **z index enable** specifies whether to use <u>Z indexing</u> for the channel.
- **z index value** specifies in **units** the value to which to reset the measurement when signal Z is high and signal A and signal B are

at the states you specify with z index phase.

z index phase specifies the states at which signal A and signal B must be while signal Z is high for NI-DAQmx to reset the measurement. If signal Z is never high while signal A and signal B are high, for example, you must choose a phase other than **A High B High**.

When signal Z transitions to high and how long it stays high varies from encoder to encoder. Refer to the documentation for the encoder to determine the timing of signal Z with respect to signal A and signal B.

A High B High (10040)	Reset the measurement when signal A and signal B are high.
A High B Low (10041)	Reset the measurement when signal A is high and signal B is low.
A Low B High (10042)	Reset the measurement when signal A is low and signal B high.
A Low B Low (10043)	Reset the measurement when signal A and signal B are low.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **pulses per revolution** is the number of pulses the encoder generates per revolution. This value is the number of pulses on either signal A or signal B, not the total number of pulses on both signal A and signal B.
- **initial angle** is the starting angle of the encoder. This value is in the units you specify with the **units** input.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **decoding type** specifies how to count and interpret the pulses the

encoder generates on signal A and signal B. **X1**, **X2**, and **X4** are valid for quadrature encoders only. **Two Pulse Counting** is valid only for two-pulse encoders.

X2 and X4 decoding are more sensitive to smaller changes in position than X1 encoding, with X4 being the most sensitive. However, more sensitive decoding is more likely to produce erroneous measurements if vibration exists in the encoder or other noise exists in the signals.

Two Pulse Counting (10313)	Increment the count on rising edges of signal A. Decrement the count on rising edges of signal B.
X1 (10090)	If signal A leads signal B, count the rising edges of signal A. If signal B leads signal A, count the falling edges of signal A.
X2 (10091)	Count the rising and falling edges of signal A.
X4 (10092)	Count the rising and falling edges of signal A and signal B.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Linear Encoder

Creates a channel that uses a linear encoder to measure linear position. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signals to the <u>default input terminals</u> of the counter unless you select different input terminals.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return linear position

measurements from the channel.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Inches (10379)	Inches.
Meters (10219)	Meters.
Ticks (10304)	Timebase ticks.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **z index enable** specifies whether to use <u>Z indexing</u> for the channel.
- **z index value** specifies in **units** the value to which to reset the measurement when signal Z is high and signal A and signal B are

at the states you specify with z index phase.

z index phase specifies the states at which signal A and signal B must be while signal Z is high for NI-DAQmx to reset the measurement. If signal Z is never high while signal A and signal B are high, for example, you must choose a phase other than **A High B High**.

When signal Z transitions to high and how long it stays high varies from encoder to encoder. Refer to the documentation for the encoder to determine the timing of signal Z with respect to signal A and signal B.

A High B High (10040)	Reset the measurement when signal A and signal B are high.
A High B Low (10041)	Reset the measurement when signal A is high and signal B is low.
A Low B High (10042)	Reset the measurement when signal A is low and signal B high.
A Low B Low (10043)	Reset the measurement when signal A and signal B are low.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **distance per pulse** is the distance to measure for each pulse the encoder generates on signal A or signal B. This value is in the units you specify with the **units** input.
- **initial position** is the position of the encoder when you begin the measurement. This value is in the units you specify with the **units** input.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **decoding type** specifies how to count and interpret the pulses the

encoder generates on signal A and signal B. **X1**, **X2**, and **X4** are valid for quadrature encoders only. **Two Pulse Counting** is valid only for two-pulse encoders.

X2 and X4 decoding are more sensitive to smaller changes in position than X1 encoding, with X4 being the most sensitive. However, more sensitive decoding is more likely to produce erroneous measurements if vibration exists in the encoder or other noise exists in the signals.

Two Pulse Counting (10313)	Increment the count on rising edges of signal A. Decrement the count on rising edges of signal B.
X1 (10090)	If signal A leads signal B, count the rising edges of signal A. If signal B leads signal A, count the falling edges of signal A.
X2 (10091)	Count the rising and falling edges of signal A.
X4 (10092)	Count the rising and falling edges of signal A and signal B.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI Two Edge Separation

Creates a channel that <u>measures the amount of time between the rising</u> <u>or falling edge of one digital signal and the rising or falling edge of</u> <u>another digital signal</u>. You can create only one counter input channel at a time with this VI because a task can include only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signals to the <u>default input</u> <u>terminals</u> of the counter unless you select different input terminals.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return time or period

measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Seconds (10364)	Seconds.
Ticks (10304)	Timebase ticks.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the maximum value you expect to measure.
- **minimum value** specifies in **units** the minimum value you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the

name of the custom scale to this input and set **units** to **From Custom Scale**.

first edge specifies on which edge of the first signal to start each measurement.

Falling (10171)	Start each measurement on the falling edge of the first signal.
Rising (10280)	Start each measurement on the rising edge of the first signal.

second edge specifies on which edge of the second signal to stop each measurement.

Falling (10171)	Stop each measurement on the falling edge of the second signal.
Rising (10280)	Stop each measurement on the rising edge of the second signal.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CI GPS Timestamp

Creates a channel that uses a special purpose counter to <u>take a</u> <u>timestamp</u> and synchronizes that counter to a GPS receiver. You can create only one counter input channel at a time with this VI because a task can contain only one counter input channel. To read from multiple counters simultaneously, use a separate task for each counter. Connect the input signals to the <u>default input terminals</u> of the counter unless you select different input terminals.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the name of the counter to use to create the virtual channel. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return the timestamp.

Custom Scale (10065)	value, you must wire a custom scale name to the custom scale name input.
Seconds (10364)	Seconds.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **gps synchronization method** specifies the method to use to synchronize the counter to a GPS receiver.

IRIG-B (10070)	Use the IRIG-B synchronization method. The GPS receiver sends one synchronization pulse per second, as well as information about the number of days, hours, minutes, and seconds that elapsed since the beginning of the current year.
None (10230)	Do not synchronize the counter to a GPS receiver. The timestamp measurement returns the number of seconds that elapsed since the device powered up unless you set the <u>CI.Timestamp.InitialSeconds</u> property.

PPS Use the PPS synchronization method. The GPS receiver (10080) sends one synchronization pulse per second, but does not send any timing information. The timestamp measurement returns the number of seconds that elapsed since the device powered up unless you set the <u>CI.Timestamp.InitialSeconds</u> property.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

CO Pulse Freq

Creates channel(s) to <u>generate digital pulses</u> that **frequency** and **duty cycle** define. The pulses appear on the <u>default output terminal</u> of the counter unless you select a different output terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the names of the counters to use to create the virtual channels. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of counters to this input. If you have an array of counters, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units in which to define pulse frequency.

Hz (10373) Hertz.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **frequency** specifies at what frequency to generate pulses.
- **duty cycle** is the width of the pulse divided by the pulse period. NI-DAQmx uses this ratio combined with frequency to determine pulse width and the interval between pulses.
- **idle state** specifies the resting state of the output terminal.

High (10192)Terminal is at a high state at rest.Low (10214)Terminal is at a low state at rest.

- **initial delay** is the amount of time in seconds to wait before generating the first pulse.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

CO Pulse Time

Creates channel(s) to <u>generate digital pulses</u> defined by the amount of time the pulse is at a high state and the amount of time the pulse is at a low state. The pulses appear on the <u>default output terminal</u> of the counter unless you select a different output terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the names of the counters to use to create the virtual channels. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of counters to this input. If you have an array of counters, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units in which to define pulse high and low time.

Seconds (10364) Seconds.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **high time** is the amount of time the pulse is high.
- **Iow time** is the amount of time the pulse is low.
- **idle state** specifies the resting state of the output terminal.

High (10192)	Terminal is at a high state at rest.
Low (10214)	Terminal is at a low state at rest.

- **initial delay** is the amount of time in seconds to wait before generating the first pulse.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

CO Pulse Ticks

Creates channel(s) to <u>generate digital pulses</u> defined by the number of timebase ticks that the pulse is at a high state and the number of timebase ticks that the pulse is at a low state. The pulses appear on the <u>default output terminal</u> of the counter unless you select a different output terminal.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **counter** specifies the names of the counters to use to create the virtual channels. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of counters to this input. If you have an array of counters, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

source of ticks is the <u>terminal</u> to which you connect an external timebase. A <u>DAQmx terminal constant</u> lists all terminals available

on devices installed in the system. You also can specify a source terminal by wiring a string that contains a <u>terminal name</u>.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **high ticks** is the number of timebase ticks the pulse is high.
- **low ticks** is the number of timebase ticks the pulse is low.
- idle state specifies the resting state of the output terminal.

High (10192)Terminal is at a high state at rest.Low (10214)Terminal is at a low state at rest.

- **initial delay** is the number of timebase ticks to wait before generating the first pulse.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

TEDS AI Voltage

Creates channel(s) to <u>measure voltage</u>. You must configure the physical channel(s) with TEDS information to use this VI. If the measurement requires the use of internal excitation or you need excitation to scale the voltage, use the <u>TEDS AI Custom Voltage with Excitation</u> instance of this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
From TEDS (12516)	Units defined by TEDS information associated with the channel.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

	terminal configuration for the channel.
Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

TEDS AI Temp TC

Creates channel(s) that use a <u>thermocouple</u> to <u>measure temperature</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **cjc value** specifies in **units** the temperature of the <u>cold junction</u> if you set **cjc source** to **Constant Value**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **cjc source** specifies the source of <u>cold-junction compensation</u>.

Channel (10113)	You must use the cjc channel input to specify a channel.
Constant Value (10116)	You must use the cjc value input to specify the cold- junction temperature.
Internal (10200)	Use a cold-junction compensation channel built into the terminal block. If no such channel is available, this VI returns an error.

- **cjc channel** specifies the channel that acquires the temperature of the thermocouple <u>cold-junction</u> if you set **cjc source** to **Channel**. You can use a global channel or another virtual channel already in the task. If the channel is a temperature channel, NI-DAQmx acquires the temperature in the correct units. Other channel types, such as a resistance channel with a custom sensor, must use a custom scale to scale values to degrees Celsius.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

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

TEDS AI Temp RTD

Creates channel(s) that use an <u>RTD</u> to <u>measure temperature</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **resistance configuration** specifies the number of wires to use for

resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

current excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

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

TEDS AI Thermistor lex

Creates channel(s) that use a <u>thermistor</u> to <u>measure temperature</u>. Use this instance when the thermistor requires current excitation. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.

resistance configuration specifies the number of wires to use for resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

current excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

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

TEDS AI Thermistor Vex

Creates channel(s) that use a <u>thermistor</u> to <u>measure temperature</u>. Use this instance when the thermistor requires voltage excitation. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return temperature measurements.

deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
K (10325)	Kelvins.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.

resistance configuration specifies the number of wires to use for resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.
4-Wire (4)	4-Wire mode.

voltage excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **r1** specifies in ohms the value of the reference resistor.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- voltage excitation value specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

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

TEDS AI Current

Creates channel(s) to <u>measure current</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
From TEDS (12516)	Units defined by TEDS information associated with the channel.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.

Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential <u>Pseudodifferential mode</u> . (12529)	
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **shunt resistor location** specifies the location of the shunt resistor. For devices with built-in shunt resistors, specify the location as **Internal**. For devices that do not have built-in shunt resistors, you must attach an external one, set this input to **External** and use the **external shunt resistor value** input to specify the value of the resistor.

default (-1)	NI-DAQmx selects Internal for devices with an internal shunt resistor. NI-DAQmx selects External for devices without an internal shunt resistor.
External (10167)	Use a shunt resistor external to the device. You must use the external shunt resistor value input to specify the value of the shunt resistor.
Internal (10200)	Use the built-in shunt resistor of the device.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- external shunt resistor value specifies in ohms the resistance of an external shunt resistor.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out

indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

TEDS AI Resistance

Creates channel(s) to <u>measure resistance</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
From TEDS (12516)	Units defined by TEDS information associated with the channel.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- resistance configuration specifies the number of wires to use for resistive measurements.

2-Wire (2)	2-Wire mode.
3-Wire (3)	3-Wire mode.

4-Wire (4) 4-Wire mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **current excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The

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

TEDS AI Strain Gage

Creates channel(s) to <u>measure strain</u>. You must configure the physical channel(s) with TEDS information to use this VI.



task in specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.

- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return strain measurements.

FromUnits a custom scale specifies. If you select thisCustomvalue, you must wire a custom scale name to the

Scale (10065)	custom scale name input.	
Strain (10299)	Strain.	

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **bridge information** specifies information about the bridge configuration and measurement.
 - **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **lead wire resistance** is the amount of resistance in ohms in the lead wires. Ideally, this value is the same for all leads.
- **initial bridge voltage** specifies in volts the output voltage of the bridge in the unloaded condition. NI-DAQmx subtracts this value from any measurements before applying scaling equations. Perform a voltage measurement on the bridge with no strain applied to determine this value.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

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

TEDS AI LVDT

Creates channel(s) that use an <u>LVDT</u> to <u>measure linear position</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return linear position

measurements from the channel.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Inches (10379)	Inches.
Meters (10219)	Meters.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- voltage excitation wire mode is the number of leads on the sensor. Some sensors require you to tie leads together to create a

four- or five- wire sensor. Refer to the sensor documentation for more information.

4-Wire (4)	Four wires.
5-Wire (5)	Five wires.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.	
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.	
None (10230)	Supply no excitation to the channel.	

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **voltage excitation frequency** specifies in hertz the excitation frequency that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

TEDS AI RVDT

Creates channel(s) that use an <u>RVDT</u> to measure <u>angular position</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return angular position

measurements from the channel.

Degrees (10146)	Degrees.
From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Radians (10273)	Radians.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- voltage excitation wire mode is the number of leads on the sensor. Some sensors require you to tie leads together to create a

four- or five- wire sensor. Refer to the sensor documentation for more information.

4-Wire (4)	Four wires.
5-Wire (5)	Five wires.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- voltage excitation source specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- **voltage excitation frequency** specifies in hertz the excitation frequency that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

TEDS AI Accelerometer

Creates channel(s) that use an <u>accelerometer</u> to <u>measure acceleration</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return acceleration measurements from the channel.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
g (10186)	1 g is approximately equal to 9.81 m/s/s.
m/s^2 (12470)	Meters per second per second.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.
| | terminal configuration for the channel. |
|-----------------------------------|---|
| Differential | Differential mode. |
| NRSE (10078) | Non-referenced single-ended mode. |
| Pseudodifferential (12529) | Pseudodifferential mode. |
| RSE (10083) | Referenced single-ended mode. |

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

TEDS AI Microphone

Creates channel(s) that use a microphone to <u>measure sound pressure</u>. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. You must use physical channels that you configured with TEDS information. The <u>DAQmx physical channel constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten</u> <u>Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return sound pressure

measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
Pascals (10081)	Pascals.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum sound pressure level** is the maximum instantaneous sound pressure level you expect to measure. This value is in decibels, referenced to 20 micropascals.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

default (-1)	At run time, NI-DAQmx chooses the <u>default</u> <u>terminal configuration</u> for the channel.

Differential (10106)	<u>Differential mode</u> .
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **current excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the current excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **current excitation value** specifies in amperes the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

(checkmark) to indicate a warning or that no error occurred.

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

TEDS AI Custom Voltage with Excitation

Creates channel(s) to measure voltage. Use this instance for custom sensors that require excitation. You can use the excitation to scale the measurement. You must configure the physical channel(s) with TEDS information to use this VI.



- **task in** specifies the task to which to add the virtual channels this VI creates. If you do not specify a task, NI-DAQmx creates a task for you and adds the virtual channels this VI creates to that task.
- **physical channels** specifies the names of the <u>physical channels</u> to use to create virtual channels. The <u>DAQmx physical channel</u> <u>constant</u> lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
- **name to assign** specifies a name to assign to the virtual channel this VI creates. If you do not wire a value to this input, NI-DAQmx uses the physical channel name as the virtual channel name. If you use this input to provide your own names for the virtual channels, you must use the names when you refer to these channels in other NI-DAQmx VIs and Property Nodes, such as the **source** input of the <u>DAQmx Trigger</u> VI.

If you create multiple virtual channels with one DAQmx Create Virtual Channel VI, you can specify a comma-separated list of names to assign to the virtual channels. If you provide fewer names than the number of virtual channels you create, NI-DAQmx <u>automatically assigns names</u> to the virtual channels.

units specifies the units to use to return measurements.

From Custom Scale (10065)	Units a custom scale specifies. If you select this value, you must wire a custom scale name to the custom scale name input.
From TEDS (12516)	Units defined by TEDS information associated with the channel.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **maximum value** specifies in **units** the <u>maximum value</u> you expect to measure.
- **minimum value** specifies in **units** the <u>minimum value</u> you expect to measure.
- **input terminal configuration** specifies the <u>input terminal</u> <u>configuration</u> for the channel.

	terminal configuration for the channel.
Differential (10106)	Differential mode.
NRSE (10078)	Non-referenced single-ended mode.
Pseudodifferential (12529)	Pseudodifferential mode.
RSE (10083)	Referenced single-ended mode.

- **custom scale name** specifies the name of a <u>custom scale</u> for the channel. If you want the channel to use a custom scale, wire the name of the custom scale to this input and set **units** to **From Custom Scale**.
- **voltage excitation source** specifies the source of excitation.

External (10167)	Use an excitation source other than the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
Internal (10200)	Use the built-in excitation source of the device. If you select this value, you must use the voltage excitation value input to specify the amount of excitation.
None (10230)	Supply no excitation to the channel.

- **task out** is a reference to the task after this VI runs. The task contains any newly created virtual channels. If you did not wire a value to **task in**, NI-DAQmx automatically creates the task this output refers to.
- **voltage excitation value** specifies in volts the amount of excitation that the sensor requires. Refer to the sensor documentation to determine this value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

DAQmx Global Channel Constant

Lists all <u>virtual channels</u> you create and save using the <u>DAQ Assistant</u>. Select **Browse** to select multiple channels. Right-click the constant and select **I/O Name Filtering** from the shortcut menu to <u>limit the channels</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Channel Property Node.

Creating and Editing Channels

You can launch the DAQ Assistant from the DAQmx global channel constant to <u>create a new virtual channel</u> or to <u>edit a saved virtual</u> <u>channel</u>.

Generating Code

You can use the DAQmx global channel constant to generate code for global channels.

DAQmx Read

Reads samples from the <u>task</u> or <u>virtual channels</u> you specify. The instances of this <u>polymorphic VI</u> specify what format of samples to return, whether to read a single sample or multiple samples at once, and whether to read from one or multiple channels.

The <u>DAQmx Read</u> properties include additional configuration options for read operations.

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

Select an instance

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

•

Analog DBL 1Chan 1Samp

Reads a single floating-point sample from a task that contains a single <u>analog input channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- data returns a sample. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog Wfm 1Chan 1Samp

Reads a waveform that contains a single sample from a task that contains a single <u>analog input channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- data returns a waveform. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D DBL 1Chan NSamp

Reads one or more floating-point samples from a task that contains a single <u>analog input channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel**specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of samples. Each element in the array corresponds to a sample from the channel. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog Wfm 1Chan NSamp

Reads a waveform from a task that contains a single <u>analog input</u> <u>channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel**specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- data returns a waveform. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D DBL NChan 1Samp

Reads a single floating-point sample from each channel in a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

data returns a 1D array of samples. Each element of the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task or to the order of the channels you specify with the <u>Channels to Read</u> property.

NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D Wfm NChan 1Samp

Reads a waveform that contains a single sample from each channel in a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

data returns a 1D array of waveforms. Each element of the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task or to the order of the channels you specify with the <u>Channels to Read</u> property.

NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D DBL NChan NSamp

Reads one or more floating-point samples from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- timeout specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of samples. Each row corresponds to a channel in the task. Each column corresponds to a sample from each channel. The order of the channels in the array corresponds to the order in which you add the channels to the task or to the order of the channels you specify with the <u>Channels to Read</u> property.

NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

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

Analog 1D Wfm NChan NSamp

Reads one or more waveforms from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of waveforms. Each element of the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task or to the order of the channels you specify with the <u>Channels to Read</u> property.

NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

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

Analog 2D I16 NChan NSamp

Reads one or more <u>unscaled</u> 16-bit signed integer samples from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- timeout specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of 16-bit signed integer samples. Each row corresponds to a channel in the task. Each column corresponds to a sample from that channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D I32 NChan NSamp

Reads one or more <u>unscaled</u> 32-bit signed integer samples from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- timeout specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of 32-bit signed integer samples. Each row corresponds to a channel in the task. Each column corresponds to a sample from that channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Analog 2D U16 NChan NSamp

Reads one or more <u>unscaled</u> 16-bit unsigned integer samples from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of 16-bit unsigned integer samples. Each row corresponds to a channel in the task. Each column corresponds to a sample from that channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D U32 NChan NSamp

Reads one or more <u>unscaled</u> 32-bit unsigned integer samples from a task that contains one or more <u>analog input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of 32-bit unsigned integer samples. Each row corresponds to a channel in the task. Each column corresponds to a sample from that channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Bool 1Line 1Point

Reads a single Boolean sample from a task that contains a <u>digital input</u> <u>channel</u> composed of a single line.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a Boolean sample.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Bool 1Chan 1Samp

Reads an array of Boolean values from a task that contains a single <u>digital input channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a 1D array of Boolean samples. Each element of the array corresponds to a digital line in the channel. The order of the lines in the array corresponds to the order in which you add the lines to the channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U8 1Chan 1Samp

Reads a single 8-bit unsigned integer sample from a task that contains a single <u>digital input channel</u>. Use an instance that reads 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

task out is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **bis** data returns an <u>8-bit unsigned integer sample</u>.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U16 1Chan 1Samp

Reads a single 16-bit unsigned integer sample from a task that contains a single <u>digital input channel</u>. Use an instance that reads 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

task out is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a <u>16-bit unsigned integer sample</u>.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U32 1Chan 1Samp

Reads a single 32-bit unsigned integer sample from a task that contains a single <u>digital input channel</u>. Use an instance that reads 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

task out is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- data returns a <u>32-bit unsigned integer sample</u>.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Wfm 1Chan 1Samp

Reads a digital waveform that contains a single sample from a task that contains a single <u>digital input channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a digital waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U8 1Chan NSamp

Reads one or more 8-bit unsigned integer samples from a task that contains a single <u>digital input channel</u>. Use an instance that reads 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of <u>8-bit unsigned integer samples</u>. Each element in the array corresponds to a sample from the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U16 1Chan NSamp

Reads one or more 16-bit unsigned integer samples from a task that contains a single <u>digital input channel</u>. Use an instance that reads 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of <u>16-bit unsigned integer samples</u>. Each element in the array corresponds to a sample from the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U32 1Chan NSamp

Reads one or more 32-bit unsigned integer samples from a task that contains a single <u>digital input channel</u>. Use an instance that reads 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of <u>32-bit unsigned integer samples</u>. Each element in the array corresponds to a sample from the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Wfm 1Chan NSamp

Reads a digital waveform from a task that contains a single <u>digital input</u> <u>channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel**specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

If the task acquires a finite number of samples and you set this input to -1, the VI waits for the task to acquire all requested samples, then reads those samples. If you set the <u>Read All</u> <u>Available Data</u> property to TRUE, the VI reads the samples currently available in the buffer and does not wait for the task to acquire all requested samples.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- data returns a digital waveform.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Bool NChan 1Samp

Reads a single Boolean sample from each channel in a task that contains one or more <u>digital input channels</u>. Each channel must contain only a single digital line.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of Boolean samples. Each element of the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D Bool NChan 1Samp

Reads a single sample that contains Boolean values from each channel in a task that contains one or more <u>digital input channels</u>. Each channel can contain multiple digital lines.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **(TFI)** data returns a 2D array of Boolean samples. Each row corresponds to a channel in the task. Each column corresponds to a line in that channel. The order of the channels in the array corresponds to the order that you add the channels to the task or to the order of the channels you specify in the <u>Channels to Read</u> property. The order of the lines in each channel corresponds to the order that you add the lines to the channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U8 NChan 1Samp

Reads a single 8-bit unsigned integer sample from each channel in a task that contains one or more <u>digital input channels</u>. Use an instance that reads 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

task out is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a 1D array of <u>8-bit unsigned integer samples</u>. Each element in the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U16 NChan 1Samp

Reads a single 16-bit unsigned integer sample from each channel in a task that contains one or more <u>digital input channels</u>. Use an instance that reads 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of <u>16-bit unsigned integer samples</u>. Each element in the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U32 NChan 1Samp

Reads a single 32-bit unsigned integer sample from each channel in a task that contains one or more <u>digital input channels</u>. Use an instance that reads 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of <u>32-bit unsigned integer samples</u>. Each element in the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Wfm NChan 1Samp

Reads a waveform that contains a single sample from each channel in a task that contains one or more <u>digital input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- task out is a reference to the task after this VI or function runs. If

you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- **data** returns a 1D array of digital waveforms that each contain a single sample. Each element in the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Digital 2D U8 NChan NSamp

Reads one or more 8-bit unsigned integer samples from a task that contains one or more <u>digital input channels</u>. Use an instance that reads 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of <u>8-bit unsigned integer samples</u>. Each row corresponds to a channel in the task. Each column corresponds to a sample from each channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D U16 NChan NSamp

Reads one or more 16-bit unsigned integer samples from a task that contains one or more <u>digital input channels</u>. Use an instance that reads 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of <u>16-bit unsigned integer samples</u>. Each row corresponds to a channel in the task. Each column corresponds to a sample from each channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D U32 NChan NSamp

Reads one or more 32-bit unsigned integer samples from a task that contains one or more <u>digital input channels</u>. Use an instance that reads 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 2D array of <u>32-bit unsigned integer samples</u>. Each row corresponds to a channel in the task. Each column corresponds to a sample from each channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Wfm NChan NSamp

Reads one or more digital waveforms from a task that contains one or more <u>digital input channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of digital waveforms. Each element in the array corresponds to a channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter DBL 1Samp

Reads a single floating-point sample from a counter task. Use an instance that reads floating-point values when NI-DAQmx scales counter samples to a floating-point value, such as for frequency and period measurement.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a sample. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter U32 1Samp

Reads a 32-bit unsigned integer sample from a counter task. Use an instance that reads 32-bit unsigned integers when NI-DAQmx returns counter samples unscaled, such as for event counting.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

task out is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- data returns a sample.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter 1D DBL NSamp

Reads one or more floating-point samples from a counter task. Use an instance that reads a floating-point value when NI-DAQmx scales counter samples to a floating-point value, such as for frequency and period measurement.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel**specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI

or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of samples. Each element in the array corresponds to a sample from the channel. NI-DAQmx scales the data to the units of the measurement, including any custom scaling you apply to the channels. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

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

Counter 1D U32 NSamp

Reads one or more 32-bit unsigned integer samples from a counter task. Use an instance that reads 32-bit unsigned integers when NI-DAQmx returns counter samples unscaled, such as for event counting.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel**specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns samples. Each element of the array corresponds to a sample from the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U8

Reads one or more <u>raw</u>, 8-bit unsigned integer samples from a task. Use this instance with devices that use 8-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **tween** data returns a 1D array of raw 8-bit unsigned integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U16

Reads one or more <u>raw</u>, 16-bit unsigned integer samples from a task. Use this instance with devices that use 16-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of raw 16-bit unsigned integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U32

Reads one or more <u>raw</u>, 32-bit unsigned integer samples from a task. Use this instance with devices that use 32-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **data** returns a 1D array of raw 32-bit unsigned integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D I8

Reads one or more <u>raw</u>, 8-bit signed integer samples from a task. Use this instance with devices that use 8-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **[18]** data returns a 1D array of raw 8-bit signed integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D I16

Reads one or more <u>raw</u>, 16-bit signed integer samples from a task. Use this instance with devices that use 16-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **table** data returns a 1D array of raw 16-bit signed integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D I32

Reads one or more <u>raw</u>, 32-bit signed integer samples from a task. Use this instance with devices that use 32-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **number of samples per channel** specifies the number of samples to read. If you leave this input unwired or set it to -1, NI-DAQmx determines how many samples to read based on if the task acquires samples continuously or acquires a finite number of samples.

If the task acquires samples continuously and you set this input to -1, this VI reads all the samples currently available in the buffer.

- **timeout** specifies the amount of time in seconds to wait for samples to become available. If the time elapses, the VI returns an error and any samples read before the timeout elapsed. The default timeout is 10 seconds. If you set **timeout** to -1, the VI waits indefinitely. If you set **timeout** to 0, the VI tries once to read the requested samples and returns an error if it is unable to.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **tax** returns a 1D array of raw 32-bit signed integer samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Read Property Node

A Property Node with the <u>DAQmx Read</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Start Task

Transitions the <u>task</u> to the running <u>state</u> to begin the measurement or generation. <u>Using this VI</u> is required for some applications and is optional for others.

If you do not use this VI, a measurement task starts automatically when the <u>DAQmx Read</u> VI runs. The **autostart** input of the <u>DAQmx Write</u> VI determines if a generation task starts automatically when the DAQmx Write VI runs.

If you do not use the DAQmx Start Task VI and the <u>DAQmx Stop Task</u> VI when you use the DAQmx Read VI or the DAQmx Write VI multiple times, such as in a loop, the task starts and stops repeatedly. Starting and stopping a task repeatedly reduces the performance of the application.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Stop Task

Stops the <u>task</u> and returns it to the <u>state</u> the task was in before the <u>DAQmx Start Task</u> VI ran or the <u>DAQmx Write</u> VI ran with the **autostart** input set to TRUE.

If you do not use the DAQmx Start Task VI and the DAQmx Stop Task VI when you use the <u>DAQmx Read</u> VI or the <u>DAQmx Write</u> VI multiple times, such as in a loop, the task starts and stops repeatedly. Starting and stopping a task repeatedly reduces the performance of the application.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Task Name Constant

Lists all <u>tasks</u> you create and save by using the <u>DAQ Assistant</u>. You cannot use this constant to select multiple tasks. Right-click the constant, and select **I/O Name Filtering** from the shortcut menu to <u>limit the tasks</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Task Property Node.
Creating and Editing Tasks

You can launch the DAQ Assistant from a DAQmx task name constant to <u>create a new task</u> or to <u>edit a saved task</u>.

Generating Code

You can use a DAQmx task name constant to generate code for tasks.

Converting to an Express VI

You can <u>convert</u> a DAQmx task name constant to a <u>DAQ Assistant</u> <u>Express VI</u>.

DAQmx Timing

Configures the number of samples to acquire or generate and creates a buffer when needed. The instances of this <u>polymorphic VI</u> correspond to the type of timing to use for the <u>task</u>.

The <u>DAQmx Timing</u> properties include all timing options included in this VI and additional timing options.

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

Select an instance

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Sample Clock

Sets the source of the <u>Sample Clock</u>, the rate of the Sample Clock, and the number of samples to acquire or generate.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **rate** specifies the <u>sampling rate</u> in samples per channel per second. If you use an external source for the Sample Clock, set this input to the maximum expected rate of that clock.
- **source** specifies the source <u>terminal</u> of the Sample Clock. Leave this input unwired to use the default onboard clock of the device.
- **active edge** specifies on which edges of Sample Clock pulses to acquire or generate samples.

Falling (10171)	Acquire or generate samples on falling edges of the Sample Clock.
Rising (10280)	Acquire or generate samples on rising edges of the Sample Clock.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

- **samples per channel** specifies the number of samples to acquire or generate for each channel in the task if **sample mode** is **Finite Samples**. If **sample mode** is **Continuous Samples**, NI-DAQmx uses this value to <u>determine the buffer size</u>.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

Handshaking

Determines the number of digital samples to acquire or generate using <u>digital handshaking</u> between the device and a peripheral device.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

ContinuousAcquire or generate samples until the DAQmx StopSamplesTask VI runs.

	(10123)	
	Finite Samples (10178)	Acquire or generate a finite number of samples.
	Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> single point sample mode is supported only for the sample clock and change detection timing types.
<u>U32</u>	samples per or generate f Samples. If uses this val	r channel specifies the number of samples to acquire for each channel in the task if sample mode is Finite sample mode is Continuous Samples , NI-DAQmx ue to <u>determine the buffer size</u> .
1/0	task out is a you wired a o DAQmx crea	reference to the task after this VI or function runs. If channel or list of channels to task/channels in , NI- ites this task automatically.
	error out con error occurre same error in status that the indicator on t	ntains error information. If error in indicates that an ed before this VI or function ran, error out contains the nformation. Otherwise, error out describes the error his VI or function produces. Right-click the error out the front panel and select Explain Error from the

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

shortcut menu for more information about the error.

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

Burst (Import Clock)

Configures when the DAQ device transfers data to a peripheral device, using an imported sample clock to control <u>burst handshake timing</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **sample clock rate** specifies in hertz the rate of the Sample Clock.
- **sample clock source** specifies the source <u>terminal</u> of the Sample Clock. Leave this input unwired to use the default onboard clock of the device.
- **sample clock active edge** specifies on which edges of Sample Clock pulses to acquire or generate samples.

Falling (10171)	Acquire or generate samples on falling edges of the Sample Clock.
Rising (10280)	Acquire or generate samples on rising edges of the Sample Clock.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

- **samples per channel** specifies the number of samples to acquire or generate for each channel in the task if **sample mode** is **Finite Samples**. If **sample mode** is **Continuous Samples**, NI-DAQmx uses this value to <u>determine the buffer size</u>.
- **pause when** specifies whether the task pauses while the trigger signal is high or low.

High (10192)Pause the task while the trigger signal is high.Low (10214)Pause the task while the trigger signal is low.

ready for transfer event active level specifies the polarity of the Ready for Transfer Event.

Active High (10095) Active high.

Active Low (10096) Active low.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Burst (Export Clock)

Configures when the DAQ device transfers data to a peripheral device, using the onboard Sample Clock of the DAQ device to control <u>burst</u> <u>handshake timing</u> and exporting that clock for use by the peripheral device.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **sample clock rate** specifies in hertz the rate of the Sample Clock.
- **sample clock output terminal** specifies the <u>terminal</u> to which to export the Sample Clock.
- **sample clock pulse polarity** specifies the polarity of the exported Sample Clock.

Active High (10095)	Active high.
Active Low (10096)	Active low.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

- samples per channel specifies the number of samples to acquire or generate for each channel in the task if sample mode is Finite Samples. If sample mode is Continuous Samples, NI-DAQmx uses this value to determine the buffer size.
- **pause when** specifies whether the task pauses while the trigger signal is high or low.

High (10192)Pause the task while the trigger signal is high.Low (10214)Pause the task while the trigger signal is low.

ready for transfer event active level specifies the polarity of the Ready for Transfer Event.

Active High (10095) Active high.

Active Low (10096) Active low.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Implicit

Sets only the number of samples to acquire or generate without specifying timing. Typically, you should use this instance when the task does not require sample timing, such as tasks that use counters for buffered frequency measurement, buffered period measurement, or pulse train generation.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

- samples per channel specifies the number of samples to acquire or generate for each channel in the task if sample mode is Finite Samples. If sample mode is Continuous Samples, NI-DAQmx uses this value to determine the buffer size.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Use Waveform

Uses the **dt** component of the **waveform** input to determine the Sample Clock rate. **dt** is the time in seconds between samples. If **sample mode** is **Finite Samples**, NI-DAQmx generates the number of samples in the waveform. This VI does not actually generate any samples. You must wire the same waveform to the <u>DAQmx Write</u> VI to generate the samples.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- waveform is the waveform to use for timing.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires or generates samples continuously or if it acquires or generates a finite number of samples.

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Change Detection

Configures the task to acquire samples on the rising and/or falling edges of the lines or ports you specify. To detect both rising and falling edges on a line or port, wire the name of that line or port to both **rising edge physical channels** and **falling edge physical channels**.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **rising edge physical channels** specifies the names of the digital lines or ports on which to detect rising edges. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in your system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input.
- **falling edge physical channels** specifies the names of the digital lines or ports on which to detect falling edges. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in your system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **sample mode** specifies if the task acquires samples continuously or if it acquires a finite number of samples.

Continuous Samples (10123)	Acquire samples until the <u>DAQmx Stop Task</u> VI runs.
Finite Samples (10178)	Acquire a finite number of samples.
Hardware Timed Single Point (12522)	Acquire samples continuously using hardware timing without a buffer. <u>Hardware timed single point</u> sample mode is supported only for the sample clock and change detection timing types.

- **samples per channel** specifies the number of samples to acquire from each channel in the task if **sample mode** is **Finite Samples**.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

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

Pipelined Sample Clock

Sets the source of the <u>Sample Clock</u>, the rate of the Sample Clock, and the number of samples to acquire or generate. The device acquires or generates samples on each Sample Clock edge, but it does not respond to certain triggers until a few Sample Clock edges later. Pipelining allows higher data transfer rates at the cost of increased trigger response latency. Refer to the device documentation for information about which triggers pipelining affects.

This timing type allows handshaking using the Pause trigger and either the Ready for Transfer event or the Data Active event. Refer to the device documentation for more information.

This timing type is supported only by the NI 6536 and NI 6537.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **rate** specifies the <u>sampling rate</u> in samples per channel per second. If you use an external source for the Sample Clock, set this input to the maximum expected rate of that clock.
- **source** specifies the source <u>terminal</u> of the Sample Clock. Leave this input unwired to use the default onboard clock of the device.
- **active edge** specifies on which edges of Sample Clock pulses to acquire or generate samples.

Falling (10171)	Acquire or generate samples on falling edges of the Sample Clock.
Rising (10280)	Acquire or generate samples on rising edges of the Sample Clock.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before

this VI or function runs, the VI or function passes the **error in** value to **error out**. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

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

Continuous Samples (10123)	Acquire or generate samples until the <u>DAQmx Stop</u> <u>Task</u> VI runs.
Finite Samples (10178)	Acquire or generate a finite number of samples.
Hardware Timed Single Point (12522)	Acquire or generate samples continuously using hardware timing without a buffer. <u>Hardware timed</u> <u>single point</u> sample mode is supported only for the sample clock and change detection timing types.

samples per channel specifies the number of samples to acquire or generate for each channel in the task if sample mode is Finite Samples. If sample mode is Continuous Samples, NI-DAQmx uses this value to <u>determine the buffer size</u>.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Timing Property Node

A Property Node with the <u>DAQmx Timing</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Trigger

Configures <u>triggering</u> for the task. The instances of this <u>polymorphic VI</u> correspond to the trigger and trigger type to configure.

The <u>DAQmx Trigger</u> properties include all triggering options included in this VI, as well as additional triggering options.

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

Select an instance

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Start None

Configures the task to start acquiring or generating samples immediately upon starting the task.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out

indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Start Digital Edge

Configures the task to start acquiring or generating samples on a rising or falling edge of a digital signal.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** specifies the name of a <u>terminal</u> where there is a digital signal to use as the source of the trigger.
- edge specifies on which edge of the digital signal to start acquiring or generating samples.

Falling (10171)	Start acquiring or generating samples on a falling edge of the digital signal.
Rising (10280)	Start acquiring or generating samples on a rising edge of the digital signal.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Start Digital Pattern

Configures a task to start acquiring or generating samples when a <u>digital</u> <u>pattern</u> is matched.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** specifies the physical channels to use for pattern matching. The order of the physical channels determines the order of the pattern. If a port is included, the order of the physical channels within the port is in ascending order.
- **pattern** specifies the <u>digital pattern</u> that must be met for the trigger to occur.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

trigger when specifies the condition under which the trigger occurs.

Pattern Does Not Match (10253)	Trigger when the physical channels do not match the specified pattern.
Pattern Matches (10254)	Trigger when the physical channels match the specified pattern.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Start Analog Edge

Configures the task to start acquiring or generating samples when an analog signal crosses the level you specify.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** is the name of a virtual channel or <u>terminal</u> where there is an analog signal to use as the source of the trigger. For E Series devices, if you use a channel name, the channel must be the first channel in the task. The only terminal you can use for E Series devices is PFI0.
- **slope** specifies on which slope of the signal to start acquiring or generating samples when the signal crosses **level**.

Falling (10171)	Start acquiring or generating samples when the signal crosses level on a falling slope.
Rising (10280)	Start acquiring or generating samples when the signal crosses level on a rising slope.

- **Ievel** specifies at what threshold to start acquiring or generating samples. Specify this value in the units of the measurement or generation. Use **slope** to specify on which slope to trigger at this threshold.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Start Analog Window

Configures the task to start acquiring or generating samples when an analog signal enters or leaves a range you specify.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** is the name of a virtual channel or <u>terminal</u> where there is an analog signal to use as the source of the trigger. For E Series devices, if you use a channel name, the channel must be the first channel in the task. The only terminal you can use for E Series devices is PFI0.
- window top is the upper limit of the window. Specify this value in the units of the measurement or generation.
- window bottom is the lower limit of the window. Specify this value in the units of the measurement or generation.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- when specifies whether the task starts measuring or generating samples when the signal enters the window or when it leaves the window. Use window bottom and window top to specify the limits of the window.

Entering Window (10163)	Start acquiring or generating samples when the signal enters the window.	
Leaving Window (10208)	Start acquiring or generating samples when the signal leaves the window.	

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Reference None

Disables reference triggering for the measurement.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

Reference Digital Edge

Configures the task to stop the acquisition when the device acquires all pretrigger samples, detects a rising or falling edge of a digital signal, and acquires all posttrigger samples. When you use a <u>Reference Trigger</u>, the default for the read <u>RelativeTo</u> property is **First Pretrigger Sample** with a read <u>Offset</u> of 0.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** specifies the name of a <u>terminal</u> where there is a digital signal to use as the source of the trigger.
- **edge** specifies on which edge of the digital signal the Reference Trigger occurs.

Falling (10171)Trigger on a falling edge of the digital signal.Rising (10280)Trigger on a rising edge of the digital signal.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **pretrigger samples per channel** specifies the minimum number of samples to acquire per channel before recognizing the Reference Trigger. The number of post-trigger samples per channel is equal to **number of samples per channel** in the <u>DAQmx Timing</u> VI minus **pretrigger samples per channel**.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Reference Digital Pattern

Configures the task to stop the acquisition when the device acquires all pretrigger samples, matches a <u>digital pattern</u>, and acquires all posttrigger samples. When you use a Reference Trigger, the default for the read RelativeTo property is First PretriggerSample with a read Offset of zero.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** specifies the physical channels to use for pattern matching. The order of the physical channels determines the order of the pattern. If a port is included, the order of the physical channels within the port is in ascending order.
- **pattern** specifies the <u>digital pattern</u> that must be met for the trigger to occur.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

trigger when specifies the condition under which the trigger occurs.

Pattern Does Not Match (10253)	Trigger when the physical channels do not match the specified pattern.	
Pattern Matches (10254)	Trigger when the physical channels match the specified pattern.	

- pretrigger samples per channel specifies the minimum number of samples to acquire per channel before recognizing the Reference Trigger. The number of post-trigger samples per channel is equal to number of samples per channel in the DAQmx Timing VI minus pretrigger samples per channel.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Reference Analog Edge

Configures the task to stop the acquisition when the device acquires all pretrigger samples; an analog signal reaches the level you specify; and the device acquires all post-trigger samples. When you use a <u>Reference</u> <u>Trigger</u>, the default for the read <u>RelativeTo</u> property is **First Pretrigger Sample** with a read <u>Offset</u> of 0.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** is the name of a virtual channel or <u>terminal</u> where there is an analog signal to use as the source of the trigger. For E Series devices, if you use a virtual channel, it must be the only channel in the task. The only terminal you can use for E Series devices is PFI0.
- **slope** specifies on which slope of the signal the Reference Trigger occurs.

Falling (10171)	Trigger when the signal crosses level on a falling slope.
Rising (10280)	Trigger when the signal crosses level on a rising slope.

- **Ievel** specifies at what threshold to trigger. Specify this value in the units of the measurement or generation. Use **slope** to specify on which slope to trigger at this threshold.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error

out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **pretrigger samples per channel** specifies the minimum number of samples to acquire per channel before recognizing the Reference Trigger. The number of post-trigger samples per channel is equal to **number of samples per channel** in the <u>DAQmx Timing</u> VI minus **pretrigger samples per channel**.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Reference Analog Window

Configures the task to stop the acquisition when the device acquires all pretrigger samples; an analog signal enters or leaves a range you specify; and the device acquires all post-trigger samples. When you use a <u>Reference Trigger</u>, the default for the read <u>RelativeTo</u> property is **First Pretrigger Sample** with a read <u>Offset</u> of 0.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** is the name of a virtual channel or <u>terminal</u> where there is an analog signal to use as the source of the trigger. For E Series devices, if you use a virtual channel, it must be the only channel in the task. The only terminal you can use for E Series devices is PFI0.
- window top is the upper limit of the window. Specify this value in the units of the measurement or generation.
- **window bottom** is the lower limit of the window. Specify this value in the units of the measurement or generation.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- when specifies whether the Reference Trigger occurs when the signal enters the window or when it leaves the window. Use window bottom and window top to specify the limits of the window.

Entering Window (10163)	Trigger when the signal enters the window.
Leaving Window (10208)	Trigger when the signal leaves the window.

- **pretrigger samples per channel** specifies the minimum number of samples to acquire per channel before recognizing the Reference Trigger. The number of post-trigger samples per channel is equal to **number of samples per channel** in the DAQmx Timing VI minus **pretrigger samples per channel**.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

Advance None

Disables advance triggering for the task.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

Advance Digital Edge

Configures a switch task to advance to the next entry in a scan list on a rising or falling edge of a digital signal.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **source** specifies the name of a <u>terminal</u> where there is a digital signal to use as the source of the trigger.
- **edge** specifies on which edge of a digital signal to advance to the next entry in the scan list.

Falling (10171)	Advance to the next scan list entry on a falling edge of the digital signal.
Rising (10280)	Advance to the next scan list entry on a rising edge of the digital signal.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Trigger Property Node

A Property Node with the <u>DAQmx Trigger</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Wait Until Done

Waits for the measurement or generation to complete. Use this VI to <u>ensure that the specified operation is complete</u> before you stop the task.

task/channels in ~~~~	DAQmx task out
timeout (sec)	
error in 🚥	error out

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- timeout (sec) specifies the maximum amount of time in seconds to wait for the measurement or generation to complete. This VI returns an error if the time elapses. The default is 10. If you set timeout (sec) to -1, the VI waits indefinitely. If you set timeout (sec) to 0, the VI checks once and returns an error if the measurement or generation is not done.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Write

Writes samples to the <u>task</u> or <u>virtual channels</u> you specify. The instances of this <u>polymorphic VI</u> specify the format of the samples to write, whether to write one or multiple samples, and whether to write to one or multiple channels.

If the task uses on-demand timing, this VI returns only after the device generates all samples. On-demand is the default timing type if you do not use the <u>DAQmx Timing</u> VI. If the task uses any timing type other than ondemand, this VI returns immediately and does not wait for the device to generate all samples. Your application must determine if the task is done to ensure that the device generated all samples.

The <u>DAQmx Write</u> properties include additional configuration options for write operations.

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

Select an instance

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

-

Analog DBL 1Chan 1Samp

Writes a floating-point sample to a <u>task</u> that contains a single <u>analog</u> <u>output channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a sample to write to the task. The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog Wfm 1Chan 1Samp

Writes a waveform that contains a single sample to a <u>task</u> that contains a single <u>analog output channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a waveform to write to the task. The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D DBL 1Chan NSamp

Writes one or more floating-point samples to a <u>task</u> that contains a single <u>analog output channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of samples to write to the task. Each element of the array corresponds to a sample to write.

The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

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

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog Wfm 1Chan NSamp

Writes a waveform to a task that contains a single analog output channel.

auto start task/channels in data error in error in error out data

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a waveform to write to the task. The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ</u> <u>Assistant</u> to specify these units.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D DBL NChan 1Samp

Writes a single floating-point sample to each channel in a <u>task</u> that contains one or more <u>analog output channels</u>.



TF

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of samples to write to the task. Each element of the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task.

The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 1D Wfm NChan 1Samp

Writes a waveform that contains a single sample to each channel in a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of waveforms to write to the task. Each element in the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task.

The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

TF

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D DBL NChan NSamp

Writes one or more floating-point samples to a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of samples to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel. The order of the channels in the array corresponds to the order in which you add the channels to the task.

The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Analog 1D Wfm NChan NSamp

Writes one or more waveforms to a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of waveforms to write to the task. Each element in the array corresponds to a channel in the task. The order of the channels in the array corresponds to the order in which you add the channels to the task.

The data you write must be in the units of the generation, including any custom scales. Use the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQ Assistant</u> to specify these units.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

TF

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D I16 NChan NSamp

Writes one or more <u>unscaled</u>, 16-bit signed integer samples to a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of 16-bit signed integer samples to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D I32 NChan NSamp

Writes one or more <u>unscaled</u>, 32-bit signed integer samples to a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of 32-bit signed integer samples to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog 2D U16 NChan NSamp

Writes one or more <u>unscaled</u> 16-bit unsigned integer samples to a <u>task</u> that contains one or more <u>analog output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of 16-bit unsigned integer samples to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Bool 1Line 1Point

Writes a single Boolean sample to a <u>task</u> that contains a <u>digital output</u> <u>channel</u> composed of a single line.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- data contains a Boolean sample to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Bool 1Chan 1Samp

Writes a single sample of Boolean values to a <u>task</u> that contains a single <u>digital output channel</u>. The channel can contain one or more digital lines.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of Boolean values to write to the task. Each element of the array corresponds to a digital line within the channel. The order of the lines in the array corresponds to the order in which you add the lines to the channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U8 1Chan 1Samp

Writes a single 8-bit unsigned integer sample to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains an <u>8-bit unsigned integer sample</u> to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U16 1Chan 1Samp

Writes a single 16-bit unsigned integer sample to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains an <u>16-bit unsigned integer sample</u> to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital U32 1Chan 1Samp

Writes a single 32-bit unsigned integer sample to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a <u>32-bit unsigned integer sample</u> to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Wfm 1Chan 1Samp

Writes a digital waveform that contains a single sample to a <u>task</u> that contains a single <u>digital output channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- data contains a digital waveform to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U8 1Chan NSamp

Writes one or more 8-bit unsigned integer samples to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>8-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a point of data to write.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U16 1Chan NSamp

Writes one or more 16-bit unsigned integer samples to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>16-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a point of data to write to the channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U32 1Chan NSamp

Writes one or more 32-bit unsigned integer samples to a <u>task</u> that contains a single <u>digital output channel</u>. Use an instance that writes 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>32-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a point of data to write to the channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Wfm 1Chan NSamp

Writes a digital waveform to a <u>task</u> that contains a single <u>digital output</u> <u>channel</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- data contains a digital waveform to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Bool NChan 1Samp

Writes a single sample of Boolean values to each channel in a <u>task</u> that contains multiple <u>digital output channels</u>. Each channel in the task can contain only one digital line.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of Boolean samples to write to the task. Each element of the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D Bool NChan 1Samp

Writes a single sample of Boolean values to each channel in a <u>task</u> that contains one or more <u>digital output channels</u>. The channels can contain one or more digital lines.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **ITF** data contains a 2D array of Boolean samples to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a line in that channel. The order of the channels in the array corresponds to the order in which you add the channels to the task. The order of the lines in the array corresponds to the order in which you add the lines to the channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U8 NChan 1Samp

Writes a single 8-bit unsigned integer sample to each channel in a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 8-bit unsigned integers for devices with up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>8-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D U16 NChan 1Samp

Writes a single 16-bit unsigned integer sample to a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>16-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Digital 1D U32 NChan 1Samp

Writes a single 32-bit unsigned integer sample to a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of <u>32-bit unsigned integer samples</u> to write to the task. Each element in the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Wfm NChan 1Samp

Writes a digital waveform that contains a single sample to each channel in a <u>task</u> that contains one or more <u>digital output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of waveforms to write to the task. Each element in the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D U8 NChan NSamp

Writes one or more 8-bit unsigned integer samples to a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 8-bit unsigned integers for devices that have up to eight lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of <u>8-bit unsigned integer samples</u> to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D U16 NChan NSamp

Writes one or more 16-bit unsigned integer samples to a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 16-bit unsigned integers for devices with up to 16 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of <u>16-bit unsigned integer samples</u> to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 2D U32 NChan NSamp

Writes one or more 32-bit unsigned integer samples to a <u>task</u> that contains one or more <u>digital output channels</u>. Use an instance that writes 32-bit unsigned integers for devices with up to 32 lines per port.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 2D array of <u>32-bit unsigned integer samples</u> to write to the task. Each row corresponds to a channel in the task. Each column corresponds to a sample to write to each channel.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital 1D Wfm NChan NSamp

Writes one or more digital waveforms to a <u>task</u> that contains one or more <u>digital output channels</u>.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- data contains a 1D array of waveforms to write to the task. Each element in the array corresponds to a channel in the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Freq 1Chan 1Samp

Writes a new pulse frequency and duty cycle to a continuous counter output task that contains a single channel.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **frequency** specifies at what frequency to generate pulses.
- **duty cycle** is the width of the pulse divided by the pulse period. NI-DAQmx uses this ratio combined with frequency to determine pulse width and the interval between pulses.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Time 1Chan 1Samp

Writes a new pulse high time and low time to a continuous counter output task that contains a single channel.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **high time** is the amount of time the pulse is high.
- **Iow time** is the amount of time the pulse is low.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Ticks 1Chan 1Samp

Writes a new pulse high tick count and low tick count to a continuous counter output task that contains a single channel.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **high ticks** is the number of timebase ticks the pulse is high.
- **Iow ticks** is the number of timebase ticks the pulse is low.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Freq NChan 1Samp

Writes a new pulse frequency and duty cycle to each channel in a continuous counter output task that contains one or more channels.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains the pulse frequency and duty cycle to write. Each element of the array corresponds to a channel in the task.
 - **frequency** specifies at what frequency to generate pulses.
 - **duty cycle** is the width of the pulse divided by the pulse period. NI-DAQmx uses this ratio combined with frequency to determine pulse width and the interval between pulses.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

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

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Time NChan 1Samp

Writes a new pulse high time and low time to each channel in a continuous counter output task that contains one or more channels.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains the pulse high time and low time to write to the task. Each element of the array corresponds to a channel in the task.
 - **high time** is the amount of time the pulse is high.

Iow time is the amount of time the pulse is low.

timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Counter Ticks NChan 1Samp

Writes new pulse high tick counts and low tick counts to each channel in a continuous counter output task that contains one or more channels.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- data contains the pulse high ticks and low ticks to write to the task.
 Each element of the array corresponds to a channel in the task.
 high ticks is the number of timebase ticks the pulse is high.

Iow ticks is the number of timebase ticks the pulse is low.

- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- number of samples written per channel is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U8

Writes multiple <u>raw</u>, 8-bit unsigned integer samples to a <u>task</u>. Use this instance for devices that use 8-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U16

Writes multiple <u>raw</u>, 16-bit unsigned integer samples to a <u>task</u>. Use this instance for devices that use 16-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D U32

Writes multiple <u>raw</u>, 32-bit unsigned integer samples to a <u>task</u>. Use this instance for devices that use 32-bit unsigned integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D 18

Writes multiple <u>raw</u>, 8-bit signed integer samples to a <u>task</u>. Use this instance for devices that use 8-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D |16

Writes multiple <u>raw</u>, 16-bit signed integer samples to a <u>task</u>. Use this instance for devices that use 16-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Raw 1D I32

Writes multiple <u>raw</u>, 32-bit signed integer samples to a <u>task</u>. Use this instance for devices that use 32-bit signed integers as the internal representation of a sample.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **data** contains a 1D array of raw samples to write to the task.
- timeout specifies the amount of time in seconds to wait for the VI to write all samples. NI-DAQmx performs a timeout check only if the VI must wait before it writes data. This VI returns an error if the time elapses. The default timeout is 10 seconds. If you set timeout to -1, the VI waits indefinitely. If you set timeout to 0, the VI tries once to write the submitted samples. If the VI could not write all the submitted samples, it returns an error and the number of samples successfully written in the number of samples written per channel output.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **auto start** specifies if this VI automatically starts the task if you did not explicitly start it with the <u>DAQmx Start Task</u> VI.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **number of samples written per channel** is the actual number of samples this VI successfully wrote to each channel in the task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
DAQmx Write Property Node

A Property Node with the <u>DAQmx Write</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Advanced VIs and Functions

Owning Palette: DAQmx - Data Acquisition VIs and Functions

Use the DAQmx Advanced VIs and functions to access advanced and miscellaneous features of NI-DAQmx.

Subpalette	Description
DAQmx Calibration VIs and Functions	Use the DAQmx Calibration VIs and functions to perform internal and external calibration of devices.
DAQmx Constants & Property Nodes	The DAQmx Constants & Property Nodes palette contains all I/O constants and Property Nodes you can use in NI-DAQmx.
DAQmx Scale Setup VIs and Functions	Use the DAQmx Scale Setup VIs and Property Node to create and configure <u>custom scales</u> .
DAQmx Signal Routing VIs	Use the DAQmx Signal Routing VIs to route signals from one terminal to another.
DAQmx Events VIs and Functions	Use the DAQmx Events VIs and functions for event- driven programming using NI-DAQmx.
<u>DAQmx</u> System Setup VI and Functions	Use the DAQmx System Setup VI and functions to set up and query information about devices and software configuration.
DAQmx TEDS VIs and Function	Use the DAQmx TEDS VIs and function to configure physical channels to use and to retrieve TEDS information from a physical channel.
DAQmx Utilities VIs	Use the DAQmx Utilities VIs as helper VIs to make NI- DAQmx programming easier.

DAQmx Calibration VIs and Functions

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx Calibration VIs and functions to perform internal and external calibration of devices.

Palette Object	Description
DAQmx Adjust AO- Series Calibration	Adjusts the <u>external calibration</u> constants for an AO Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with reference voltage .
DAQmx Adjust E- Series Calibration	Adjusts the <u>external calibration</u> constants for an E Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with reference voltage .
DAQmx Adjust M- Series Calibration	Adjusts the <u>external calibration</u> constants for an M Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with reference voltage .
DAQmx Adjust SC Baseboard Calibration	Adjusts the <u>external calibration</u> constants for the baseboard of an SC Series device. You must connect a known voltage to the device and specify that voltage with reference voltage .
DAQmx Calibration Info Property Node	A Property Node with the <u>DAQmx Calibration Info</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Change External Calibration Password	Changes the external calibration password of the device.
DAQmx Close	Closes an open <u>external calibration</u> session.

External Calibration	
DAQmx Initialize External Calibration	Starts an <u>external calibration</u> session on a device.
DAQmx Perform Bridge Offset Nulling Calibration	Performs a bridge offset nulling calibration on the channels in the task. If the task measures both bridge-based sensors and non-bridge-based sensors, use the channels input to specify the names of the channels that measure bridge- based sensors.
DAQmx Perform Shunt Calibration	Performs shunt calibration for the specified channels of the task. The instances of this polymorphic VI correspond to the type of bridge sensor. Refer to the calibration procedure for your module for detailed calibration instructions.
DAQmx Restore Last External Calibration Constants	Sets the <u>self calibration</u> constants of the device to the <u>external calibration</u> constants. NI sets the external calibration constants at the factory, and those constants remain in effect until you perform a new external calibration on the device.
<u>DAQmx</u> <u>Self</u> Calibrate	Measures the onboard reference voltage of the device and adjusts the <u>self calibration</u> constants to account for any errors caused by short-term fluctuations in the operating environment. When you self calibrate a device, no external signal connections are necessary.

Subpalette	Description
DAQmx	Use the DAQmx 42xx Calibration VIs to calibrate NI 42xx
<u>42XX</u>	devices.
VIS	
<u>DAQmx</u>	Use the DAQmx DSA Calibration VIs to perform external
<u>DSA</u>	calibrations of the analog input section, analog output
Calibration	section, and timebase of DSA devices.

<u>VIs</u>	
<u>DAQmx</u>	Use the DAQmx SCXI Calibration VIs to perform external
<u>SCXI</u>	calibrations of SCXI modules.
Calibration	
<u>VIs</u>	

DAQmx 42xx Calibration VIs

Owning Palette: DAQmx Calibration VIs and Functions

Use the DAQmx 42xx Calibration VIs to calibrate NI 42xx devices.

Palette Object	Description
DAQmx Adjust 4204 Calibration	Adjusts the <u>internal and external calibration</u> constants for an NI 4204 device. You must connect a known voltage to the device and specify that voltage with reference voltage .
DAQmx Adjust 4220 Calibration	Adjusts the internal and external calibration constants for an NI 4220 device. This device requires reference signals of 0.0 volts at gains of 1, 15, 20, and 310 on a particular channel in order to perform an offset calibration for that channel. If you do not manually supply those reference signals, the device measures them internally with sample and hold circuitry enabled.
DAQmx Adjust 4224 Calibration	Adjusts the <u>internal and external calibration</u> constants for an NI 4224 device. You must connect a known voltage to the device and specify that voltage with reference voltage .

DAQmx Adjust 4204 Calibration

Adjusts the <u>internal and external calibration</u> constants for an NI 4204 device. You must connect a known voltage to the device and specify that voltage with **reference voltage**.



- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channels** specifies the physical channel(s) to calibrate.
- **Iowpass filter cutoff frequency** specifies in Hz which lowpass filter cutoff frequency to calibrate.
- **sample and hold enable** specifies whether to calibrate the channel(s) with the sample and hold circuitry of the channel enabled.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Adjust 4220 Calibration

Adjusts the <u>internal and external calibration</u> constants for an NI 4220 device. This device requires reference signals of 0.0 volts at gains of 1, 15, 20, and 310 on a particular channel in order to perform an offset calibration for that channel. If you do not manually supply those reference signals, the device measures them internally with sample and hold circuitry enabled.



- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channels** specifies the physical channel(s) to calibrate.
- **gain** specifies the gain setting to calibrate.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Adjust 4224 Calibration

Adjusts the <u>internal and external calibration</u> constants for an NI 4224 device. You must connect a known voltage to the device and specify that voltage with **reference voltage**.



- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channels** specifies the physical channel(s) to calibrate.
- **gain** specifies the gain setting to calibrate.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Adjust AO-Series Calibration

Adjusts the <u>external calibration</u> constants for an AO Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Adjust E-Series Calibration

Adjusts the <u>external calibration</u> constants for an E Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Adjust M-Series Calibration

Adjusts the <u>external calibration</u> constants for an M Series device. You must <u>connect a known voltage to the device</u> and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Adjust S-Series Calibration

Adjusts the <u>external calibration</u> constants for an S Series device. You must connect a known voltage to the device and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Adjust SC Baseboard Calibration

Adjusts the <u>external calibration</u> constants for the baseboard of an SC Series device. You must connect a known voltage to the device and specify that voltage with **reference voltage**.

Use the <u>DAQmx 42xx Calibration VIs</u> to calibrate the full signal path of NI 42xx devices after using this VI.



- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Calibration Info Property Node

A Property Node with the <u>DAQmx Calibration Info</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Change External Calibration Password

Changes the external calibration password of the device.



- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- **password** is the current calibration password for the device. This password is case sensitive. The default password for SCXI-15xx devices is SCXI. The default password for all other NI products is NI.
- **newpassword** is the new password for the device. The password can be no longer than four characters.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Close External Calibration

Closes an open external calibration session.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **action** specifies how to close the calibration session.

cancel (1)	Closes the session without saving any calibration changes.
commit (0)	Saves the calibration changes you made in the session.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx DSA Calibration VIs

Owning Palette: DAQmx Calibration VIs and Functions

Use the DAQmx DSA Calibration VIs to perform external calibrations of the analog input section, analog output section, and timebase of DSA devices.

Palette Object	Description
DAQmx Adjust DSA AI Calibration	Adjusts the <u>external calibration</u> constants for the analog input section of a DSA device. You must connect a known voltage to the device and specify that voltage with reference voltage .
DAQmx Adjust DSA AO Calibration	Adjusts the <u>external calibration</u> constants for the analog output section of a DSA device. You must use the device to generate a high voltage and low voltage at a specified gain, measure the high and low voltages, then specify the requested high voltage, low voltage, and gain along with the actual high voltage and low voltage.
DAQmx Adjust DSA Timebase Calibration	Adjusts the <u>external calibration</u> constant for the timebase of a DSA device with an adjustable oscillator. You must connect a sinusoidal signal with a known frequency to the device and specify that frequency with reference frequency .

DAQmx Adjust DSA AI Calibration

Adjusts the <u>external calibration</u> constants for the analog input section of a DSA device. You must connect a known voltage to the device and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- reference voltage specifies in volts the known voltage to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Adjust DSA AO Calibration

Adjusts the <u>external calibration</u> constants for the analog output section of a DSA device. You must use the device to generate a high voltage and low voltage at a specified gain, measure the high and low voltages, then specify the requested high voltage, low voltage, and gain along with the actual high voltage and low voltage.



■ Place on the block diagram. ■ Find on the **Functions** palette.

- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **channel** is the number of the channel to calibrate. This number is the numeric portion of the physical channel name, not the full physical channel name.
- requested low voltage is the low voltage you attempted to generate at the gain setting you specified.
- **actual low voltage** is the actual low voltage an external sensor measures.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- requested high voltage is the high voltage you attempted to generate at the gain setting you specified.
- **actual high voltage** is the actual high voltage an external sensor measures.
- **gain setting** is the gain setting you used when you attempted to generate the **requested high voltage** and **requested low voltage**.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Adjust DSA Timebase Calibration

Adjusts the <u>external calibration</u> constant for the timebase of a DSA device with an adjustable oscillator. You must connect a sinusoidal signal with a known frequency to the device and specify that frequency with **reference frequency**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference frequency** specifies in hertz the frequency of the signal to use as a reference for calibration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Starts an external calibration session on a device.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- **password** is the current calibration password for the device. This password is case sensitive. The default password for SCXI-15xx devices is SCXI. The default password for all other NI products is NI.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Perform Bridge Offset Nulling Calibration

Performs a bridge offset nulling calibration on the channels in the task. If the task measures both bridge-based sensors and non-bridge-based sensors, use the **channels** input to specify the names of the channels that measure bridge-based sensors.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **channels** is a subset of virtual channels in the task that you want to calibrate. Use this input if you do not want to calibrate all the channels in the task or if some channels in the task measure non-bridge-based sensors. If the input is empty, this VI attempts to calibrate all virtual channels in the task.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

- **skip unsupported channels** specifies whether or not to skip channels that do not support calibration. If **skip unsupported channels** is TRUE, this VI calibrates only supported channels. If FALSE, this VI calibrates the channels specified by **channels**. The default is FALSE.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Perform Shunt Calibration

Performs shunt calibration for the specified channels of the task. The instances of this polymorphic VI correspond to the type of bridge sensor. Refer to the calibration procedure for your module for detailed calibration instructions.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Strain

Performs shunt calibration for the specified channels using a strain gage sensor.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **channels** is a subset of virtual channels in the task that you want to calibrate. Use this input if you do not want to calibrate all the channels in the task or if some channels in the task measure nonbridge-based sensors. If the input is empty, this VI attempts to calibrate all virtual channels in the task.
- **shunt resistance** specifies the shunt resistance in ohms.
- **shunt element location** specifies the location of the shunt resistor.



R1 (12465)	Between V_{ch} and V_{ex+} .
R2 (12466)	Between V_{ch} and V_{ex} .
R3 (12467)	Between V_{ch+} and V_{ex-} .
R4 (14813)	Between V_{ch+} and V_{ex+} .
None (10230)	No shunt resistor connected.

error in describes error conditions that occur before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **skip unsupported channels** specifies whether or not to skip channels that do not support calibration. If **skip unsupported channels** is TRUE, this VI calibrates only supported channels. If FALSE, this VI calibrates the channels specified by **channels**. The default is FALSE.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

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

Bridge

Performs shunt calibration for the specified channels using a bridge sensor.



- **task/channels in** is the name of the task or a list of virtual channels to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- 1/01 channels is a subset of virtual channels in the task that you want to calibrate. Use this input if you do not want to calibrate all the channels in the task or if some channels in the task measure nonbridge-based sensors. If the input is empty, this VI attempts to calibrate all virtual channels in the task.
- DBL **shunt resistance** specifies the shunt resistance in ohms.
- 132 shunt element location specifies the location of the shunt resistor.



R1 (12465)	Between V_{ch} and V_{ex+} .
R2 (12466)	Between V_{ch} and V_{ex} .
R3 (12467)	Between V_{ch+} and V_{ex-} .
R4 (14813)	Between V_{ch+} and V_{ex+} .
None (10230)	No shunt resistor connected.

error in describes error conditions that occur before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **bridge resistance** specifies the bridge resistance in ohms.
- **skip unsupported channels** specifies whether or not to skip channels that do not support calibration. If **skip unsupported channels** is TRUE, this VI calibrates only supported channels. If FALSE, this VI calibrates the channels specified by **channels**. The default is FALSE.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

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

DAQmx Restore Last External Calibration Constants

Sets the <u>self calibration</u> constants of the device to the <u>external calibration</u> constants. NI sets the external calibration constants at the factory, and those constants remain in effect until you perform a new external calibration on the device.

This VI nullifies any self calibration you perform on the device. If you have never performed a self calibration on the device, this VI has no effect.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Self Calibrate

Measures the onboard reference voltage of the device and adjusts the <u>self calibration</u> constants to account for any errors caused by short-term fluctuations in the operating environment. When you self calibrate a device, no external signal connections are necessary.

For 42xx devices, this VI performs a self-calibration on the 4200 baseboard, then performs an offset-only calibration on the full signal path of the device.

device in Constant DROmx
error in error out

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx SCXI Calibration VIs

Owning Palette: DAQmx Calibration VIs and Functions

Use the DAQmx SCXI Calibration VIs to perform external calibrations of SCXI modules.

Palette Object	Description
DAQmx Adjust SCXI Calibration	Adjusts the external calibration constants for an SCXI module. The instances of this <u>polymorphic VI</u> correspond to the SCXI module you want to calibrate. Refer to the <u>calibration procedure for your module</u> for detailed calibration instructions.
DAQmx Setup SCXI Calibration	Sets up calibration for SCXI modules. The instances of this <u>polymorphic VI</u> correspond to the SCXI module you want to calibrate. Refer to the <u>calibration procedure for your module</u> for detailed calibration instructions.

DAQmx Adjust SCXI Calibration

Adjusts the external calibration constants for an SCXI module. The instances of this <u>polymorphic VI</u> correspond to the SCXI module you want to calibrate. Refer to the <u>calibration procedure for your module</u> for detailed calibration instructions.



Note After you commit the calibration, you cannot undo it. If any calibration settings are incorrect, you must recalibrate the module.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Adjusts the external calibration constants for an SCXI-1102 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1104 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1112 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1122 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1124 module. You must use the <u>DAQmx Setup SCXI Calibration</u> VI to generate a voltage or current. Measure that voltage or current, and specify the measured value with **measured output**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- measured output specifies in volts or amperes the voltage or current measured at the output channel specified in the DAQmx Setup SCXI Calibration VI.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Adjusts the external calibration constants for an SCXI-1125 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1126 module. You must connect a sinusoidal signal with a known frequency to the device and specify that frequency with **reference frequency**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference frequency** specifies in hertz the frequency of the signal to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1141 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1142 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1143 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1502 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1503 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

1503 Current

Adjusts the current calibration constants for an SCXI-1503 module. You must measure the current generated on the **physical channel** you specify and specify the measured current in **measured output**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **measured current** specifies in amperes the current measured at the specified physical channel.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Adjusts the external calibration constants for an SCXI-1520 module. This module requires reference signals of 0.0 V at gains of 1, 15, 20, and 310 on a particular channel to perform an offset calibration for that channel.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1521 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

153x

Adjusts the external calibration constants for an SCXI-1530 or SCXI-1531 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Adjusts the external calibration constants for an SCXI-1540 module. You must connect a known voltage to the module and specify that voltage with **reference voltage**.



calhandle in is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.

input calibration source specifies the input source selection.

Loopback0	Loopback the internal excitation voltage with 0 degree phase shift.
Loopback180	Loopback the internal excitation voltage with 180 degree phase shift.
Ground	Connect the channel to ground.

- **reference voltage** specifies in volts the known voltage to use as a reference for calibration.
- **measured output** specifies in volts the voltage measured at the output of the module.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status**

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Setup SCXI Calibration

Sets up calibration for SCXI modules. The instances of this <u>polymorphic</u> \underline{VI} correspond to the SCXI module you want to calibrate. Refer to the <u>calibration procedure for your module</u> for detailed calibration instructions.



Note After you commit the calibration, you cannot undo it. If any calibration settings are incorrect, you must recalibrate the module.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Sets an SCXI-1102 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1102/B/C User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Configures the specified physical channel on an SCXI-1104 module for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1104/C User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Configures the specified physical channel on an SCXI-1112 module for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1112 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1122 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the module user manual for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Writes the specified binary value to the DAC on the specified physical channel at the specified range. Measure the voltage or current generated, then use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify the measured value and adjust calibration constants.

Note Specify at least two calibration points for each physical channel and range you calibrate. Use 0 and 4095 for voltage calibration ranges. Use 255 and 4095 for current calibration ranges.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **range** specifies the range to calibrate.

0Volts to 1Volts (14629)	0 V to 1 V.
0Volts to 5Volts (14630)	0 V to 5 V.
0Volts to 10Volts (14631)	0 V to 10 V.
-1Volts to 1Volts (14632)	-1 V to 1 V.
-5Volts to 5Volts (14633)	-5 V to 5 V.
-10Volts to 10Volts (14634)	-10 V to 10 V.
0Amps to 20mAmps (14635)	0 A to 20 mA.

- **DAC value** is the binary number to write to the DAC.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error

out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1125 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1125 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Specifies the channel and range on an SCXI-1126 module for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1126 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **upper frequency limit** is the high frequency limit in hertz which, with 0 Hz as the low frequency limit, most closely encapsulates the ranges to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

Sets an SCXI-1141 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1141/1142/1143 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1142 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1141/1142/1143 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1143 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1141/1142/1143 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
Sets the SCXI-1502 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets the SCXI-1503 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1520 module to the specified gain value for calibration. This VI disables the sample-and-hold circuitry on the module so external devices that cannot supply the appropriate sample-and-hold timing signals can measure the output. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1520 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

Configures the specified physical channel on an SCXI-1521 module for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the module user manual for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

153x

Sets an SCXI-153*x* module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1530/1531 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **gain** specifies the gain setting to calibrate.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

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

calhandle out is a reference to the calibration session. Wire this output to other external calibration VIs.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Sets an SCXI-1540 module to the specified gain value for calibration. Measure calibration input and output points by supplying reference signals to the specified physical channel and measuring the outputs. Use the <u>DAQmx Adjust SCXI Calibration</u> VI to specify each input and output point and adjust calibration constants.

Note The terminal where you can measure module output depends on the configuration of the module in MAX. National Instruments recommends cabling the module to the digitizer so that the output appears on the MCH0± pins of the rear signal connector. Refer to the *SCXI-1540 User Manual* for more information on the routing of module output.



- **calhandle in** is a reference to the calibration session that you created using the <u>DAQmx Initialize External Calibration</u> VI.
- **physical channel** specifies the physical channel to calibrate.
- **excitation RMS value** is the internal AC excitation value in volts RMS.
- excitation frequency is the internal AC excitation frequency in hertz.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **calhandle out** is a reference to the calibration session. Wire this output to other external calibration VIs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Constants & Property Nodes

Owning Palette: DAQmx Advanced VIs and Functions

The DAQmx Constants & Property Nodes palette contains all I/O constants and Property Nodes you can use in NI-DAQmx.

Palette Object	Description
<u>DAQmx</u> <u>Buffer</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Buffer</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Calibration Info Property Node	A Property Node with the <u>DAQmx Calibration Info</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Channel Property Node	A Property Node with the <u>DAQmx Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all devices installed in the system.
<u>DAQmx</u> <u>Device</u> <u>Name</u> <u>Constant</u>	Lists all devices installed in the system. Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the devices</u> that the constant displays and to limit what you can enter in the constant. If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Device Property Node.
<u>DAQmx</u> <u>Device</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Device</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the

	system or supported by all the devices installed in the system.
DAQmx Export Signal Property Node	A Property Node with the <u>DAQmx Export Signal</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Global</u> <u>Channel</u> Constant	Lists all <u>virtual channels</u> you create and save using the <u>DAQ</u> <u>Assistant</u> . Select Browse to select multiple channels. Right- click the constant and select I/O Name Filtering from the shortcut menu to <u>limit the channels</u> that the constant displays and to limit what you can enter in the constant.
DAQmx Persisted Channel Property Node	A Property Node with the <u>DAQmx Persisted Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Persisted Scale Property Node	A Property Node with the <u>DAQmx Persisted Scale</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Persisted Task Property Node	A Property Node with the <u>DAQmx Persisted Task</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Physical Channel Constant	Lists all <u>physical channels</u> on devices installed in the system. Select Browse to select multiple physical channels. Right- click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the physical channels</u> that the constant displays and to limit what you can enter in the constant.

DAQmx Physical Channel Property Node	A Property Node with the <u>DAQmx Physical Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx <u>Read</u> Property Node	A Property Node with the <u>DAQmx Read</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Real-Time Property Node	A Property Node with the <u>DAQmx Real-Time</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Scale</u> <u>Name</u> Constant	Lists all <u>custom scales</u> you create and save by using the <u>DAQ Assistant</u> . Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the scales</u> that the constant displays and to limit what you can enter in the constant.
DAQmx Scale Property Node	A Property Node with the <u>DAQmx Scale</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Switch Channel Property Node	A Property Node with the <u>DAQmx Switch Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Switch</u>	Lists all switch channels, relays, or topologies available on devices installed in the system. Select Browse to select

<u>Constant</u>	multiple switch channels or relays. Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the switch resources</u> that the constant displays and to limit what you can enter in the constant.
DAQmx Switch Device Property Node	A Property Node with the <u>DAQmx Switch Device</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Switch Scan Property Node	A Property Node with the <u>DAQmx Switch Scan</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>System</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx System</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Task</u> <u>Name</u> Constant	Lists all <u>tasks</u> you create and save by using the <u>DAQ</u> <u>Assistant</u> . You cannot use this constant to select multiple tasks. Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the tasks</u> that the constant displays and to limit what you can enter in the constant.
<u>DAQmx</u> <u>Task</u> Property Node	A Property Node with the <u>DAQmx Task</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Terminal</u> Constant	Lists all <u>terminals</u> available on devices installed in the system. Select Browse to select multiple terminals. Right- click the constant and select I/O Name Filtering from the

	shortcut menu to <u>limit the terminals</u> that the constant displays and to limit what you can enter in the constant.
DAQmx Timing Property Node	A Property Node with the <u>DAQmx Timing</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Timing Source Property Node	A Property Node with the <u>DAQmx Timing Source</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Trigger</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Trigger</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Watchdog Property Node	A Property Node with the <u>DAQmx Watchdog</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all devices installed in the system.
DAQmx Write Property Node	A Property Node with the <u>DAQmx Write</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

DAQmx Physical Channel Constant

Lists all <u>physical channels</u> on devices installed in the system. Select **Browse** to select multiple physical channels. Right-click the constant, and select **I/O Name Filtering** from the shortcut menu to <u>limit the physical channels</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and does not provide any configuration options for physical channels.

DAQmx Scale Name Constant

Lists all <u>custom scales</u> you create and save by using the <u>DAQ Assistant</u>. Right-click the constant, and select **I/O Name Filtering** from the shortcut menu to <u>limit the scales</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Scale Property Node.

Creating and Editing a Custom Scale

You can launch the DAQ Assistant from the DAQmx scale name constant to <u>create a new custom scale</u> or to <u>edit an existing custom scale</u>.

DAQmx Switch Constant

Lists all switch channels, relays, or topologies available on devices installed in the system. Select **Browse** to select multiple switch channels or relays. Right-click the constant, and select **I/O Name Filtering** from the shortcut menu to <u>limit the switch resources</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Switch Device Property Node, DAQmx Switch Channel Property Node, or DAQmx Switch Scan Property Node.

DAQmx Events VIs and Functions

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx Events VIs and functions for event-driven programming using NI-DAQmx.

Palette Object	Description
DAQmx Create Event	Contains a <u>Register for Events</u> function, an <u>Event structure</u> , and a <u>DAQmx Task Name constant</u> . You must <u>register</u> the DAQmx event you want to use and <u>configure</u> the Event structure to handle that event. You must also use the DAQmx Task Name constant to select or <u>create</u> a task. You can use a DAQmx Task Name control in place of the constant.

Subpalette	Description
<u>Events</u>	Use the Event functions to register events dynamically and
Functions	to create user events.

DAQmx Create Event (Not Available in LabVIEW 7.x)

Contains a <u>Register for Events</u> function, an <u>Event structure</u>, and a <u>DAQmx Task Name constant</u>. You must <u>register</u> the DAQmx event you want to use and <u>configure</u> the Event structure to handle that event. You must also use the DAQmx Task Name constant to select or <u>create</u> a task. You can use a DAQmx Task Name control in place of the constant.



Note You cannot configure the Event structure using the LabVIEW Base Development System.

DAQmx Scale Setup VIs and Functions

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx Scale Setup VIs and Property Node to create and configure <u>custom scales</u>.

Palette Object	Description
DAQmx Compute Reverse Polynomial Coefficients	Computes a set of coefficients for a polynomial that approximates the inverse of the polynomial with the coefficients you specify with the coefficients input. This VI generates a table of <i>x</i> versus <i>y</i> values over the range of <i>x</i> . This VI then finds a polynomial fit, using the least squares method to compute a polynomial that computes <i>x</i> when given a value for <i>y</i> .
<u>DAQmx</u> <u>Create</u> <u>Scale</u>	Creates and configures a <u>custom scale</u> . Apply custom scales to input and output channels to provide additional scaling. The instances of this <u>polymorphic VI</u> correspond to the type of scale this VI creates.
<u>DAQmx</u> <u>Scale</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Scale</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

DAQmx Compute Reverse Polynomial Coefficients

Computes a set of coefficients for a polynomial that approximates the inverse of the polynomial with the coefficients you specify with the **coefficients** input. This VI generates a table of *x* versus *y* values over the range of *x*. This VI then finds a polynomial fit, using the least squares method to compute a polynomial that computes *x* when given a value for *y*.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **coefficients** is the array of coefficients for the polynomial that computes *y* given a value of *x*. Each element of the array corresponds to a term of the equation. For example, if index three of the array is 9, the fourth term of the equation is $9x^3$.
- **maximum valid value of x** is the maximum value of *x* for which you use the polynomial. This is the largest value of *x* for which the VI generates a *y* value in the table.
- **minimum valid value of x** is the minimum value of *x* for which you use the polynomial. This is the smallest value of *x* for which the VI generates a *y* value in the table.
- **number of points to compute within x range** is the number of points in the table of *x* versus *y* values. The VI spaces the values evenly between **minimum valid value of x** and **maximum valid value of x**.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error

out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- reverse polynomial order is the order of the reverse polynomial to compute. For example, an input of 3 indicates a 3rd order polynomial. A value of -1 indicates a reverse polynomial of the same order as the forward polynomial.
- **reverse coefficients** is the array of coefficients for the reverse polynomial. Each element of the array corresponds to a term of the equation. For example, if index three of the array is 9, the fourth term of the equation is $9y^3$.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Create Scale

Creates and configures a <u>custom scale</u>. Apply custom scales to input and output channels to provide additional scaling. The instances of this <u>polymorphic VI</u> correspond to the type of scale this VI creates.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Linear

Г

Creates a <u>custom scale</u> that uses the equation y=mx+b, where x is a prescaled value, and y is a scaled value. The equation is identical for input and output. If the equation is in the form x=my+b, you must first solve for y in terms of x.



- **name** identifies the custom scale for later use, such as with the DAQmx Create Virtual Channel VI
- **Slope** is the slope, m, in the equation.
- **y-intercept** is the y-intercept, *b*, in the equation.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **pre-scaled units** is the units of the values to scale.

Τ

Amps (10342)	Amperes.
deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
Degrees (10146)	Degrees.
g (10186)	G. 1 g is approximately equal to 9.81 m/s/s.
Hz (10373)	Hertz.
Inches (10379)	Inches.
Kelvins (10325)	Kelvins.
Meters (10219)	Meters.
Ohms (10384)	Ohms.
Radians (10273)	Radians.
Seconds (10364)	Seconds.
Strain (10299)	Strain.
Volts (10348)	Volts.

- **scaled units** is the units to use for the scaled value. You can use an arbitrary string. LabVIEW uses the units to label a graph or chart.
- **scale out** is a reference to the custom scale this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

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

Map Ranges

Creates a <u>custom scale</u> that scales values proportionally from a range of pre-scaled values to a range of scaled values.



- **name** identifies the custom scale for later use, such as with the <u>DAQmx Create Virtual Channel</u> VI
- **scaled minimum** is the smallest value in the range of scaled values. NI-DAQmx maps this value to **pre-scaled minimum**. Read operations clip samples that are smaller than this value. Write operations generate errors for samples that are smaller than this value.
- **scaled maximum** is the largest value in the range of scaled values. NI-DAQmx maps this value to **pre-scaled maximum**. Read operations clip samples that are larger than this value. Write operations generate errors for samples that are larger than this value.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **pre-scaled minimum** is the smallest value in the range of prescaled values. NI-DAQmx maps this value to **scaled minimum**.
- **pre-scaled maximum** is the largest value in the range of prescaled values. NI-DAQmx maps this value to **scaled maximum**.
- **pre-scaled units** is the units of the values to scale.

Amps (10342)	Amperes.
deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
Degrees (10146)	Degrees.
g (10186)	G. 1 g is approximately equal to 9.81 m/s/s.
Hz (10373)	Hertz.
Inches (10379)	Inches.
Kelvins (10325)	Kelvins.
Meters (10219)	Meters.
Ohms (10384)	Ohms.
Radians (10273)	Radians.
Seconds (10364)	Seconds.
Strain (10299)	Strain.
Volts (10348)	Volts.

- **scaled units** is the units to use for the scaled value. You can use an arbitrary string. LabVIEW uses the units to label a graph or chart.
- **scale out** is a reference to the custom scale this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

Polynomial

Creates a <u>custom scale</u> that uses an *n*th order polynomial equation. NI-DAQmx requires both a polynomial to convert pre-scaled values to scaled values (forward) and a polynomial to convert scaled values to prescaled values (reverse). If you only know one set of coefficients, use the <u>DAQmx Compute Reverse Polynomial Coefficients</u> VI to generate the other set.



- **name** identifies the custom scale for later use, such as with the DAQmx Create Virtual Channel VI
- **forward coeff** is an array of coefficients for the polynomial that converts pre-scaled values to scaled values. Each element of the array corresponds to a term of the equation. For example, if index three of the array is 9, the fourth term of the equation is $9x^3$.
- **reverse coeff** is an array of coefficients for the polynomial that converts scaled values to pre-scaled values. Each element of the array corresponds to a term of the equation. For example, if index three of the array is 9, the fourth term of the equation is 9y³.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

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

Amps (10342)	Amperes.
deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
Degrees (10146)	Degrees.
g (10186)	G. 1 g is approximately equal to 9.81 m/s/s.
Hz (10373)	Hertz.
Inches (10379)	Inches.
Kelvins (10325)	Kelvins.
Meters (10219)	Meters.
Ohms (10384)	Ohms.
Radians (10273)	Radians.
Seconds (10364)	Seconds.
Strain (10299)	Strain.
Volts (10348)	Volts.

- **scaled units** is the units to use for the scaled value. You can use an arbitrary string. LabVIEW uses the units to label a graph or chart.
- **scale out** is a reference to the custom scale this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

Creates a <u>custom scale</u> that maps an array of pre-scaled values to an array of corresponding scaled values. NI-DAQmx applies linear interpolation to values that fall between the values in the table. Read operations clip scaled samples that are outside the maximum and minimum scaled values found in the table. Write operations generate errors for samples that are outside the minimum and maximum scaled values found in the table.



- **name** identifies the custom scale for later use, such as with the DAQmx Create Virtual Channel VI
- **scaled values** is the array of scaled values that map to the values in **pre-scaled values**.
- **pre-scaled values** is the array of pre-scaled values that map to the values in **scaled values**.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

pre-scaled units is the units of the values to scale.

Amps (10342)	Amperes.
deg C (10143)	Degrees Celsius.
deg F (10144)	Degrees Fahrenheit.
deg R (10145)	Degrees Rankine.
Degrees (10146)	Degrees.
g (10186)	G. 1 g is approximately equal to 9.81 m/s/s.
Hz (10373)	Hertz.
Inches (10379)	Inches.
Kelvins (10325)	Kelvins.
Meters (10219)	Meters.
Ohms (10384)	Ohms.
Radians (10273)	Radians.
Seconds (10364)	Seconds.
Strain (10299)	Strain.
Volts (10348)	Volts.

- **scaled units** is the units to use for the scaled value. You can use an arbitrary string. LabVIEW uses the units to label a graph or chart.
- **scale out** is a reference to the custom scale this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

code is the error or warning code. If **status** is TRUE, **code**

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

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

DAQmx Scale Property Node

A Property Node with the <u>DAQmx Scale</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Signal Routing VIs

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx Signal Routing VIs to route signals from one terminal to another.

Palette Object	Description
<u>DAQmx</u> <u>Connect</u> <u>Terminals</u>	Creates a route between a source and destination <u>terminal</u> . The route can carry a variety of digital signals, such as triggers, clocks, and hardware events.
DAQmx Disconnect Terminals	Removes signal routes you created by using the <u>DAQmx</u> <u>Connect Terminals</u> VI. The DAQmx Disconnect Terminals VI cannot remove task-based routes, such as those you create through timing and triggering configuration.
<u>DAQmx</u> <u>Tristate</u> <u>Output</u> <u>Terminal</u>	Sets a terminal to <u>high-impedance state</u> . If you connect an external signal to a terminal on the I/O connector, the terminal must be in high-impedance state. Otherwise, the device could double-drive the terminal and damage the hardware. If you use this VI on a terminal in an active route, the VI fails and returns an error.

DAQmx Connect Terminals

Creates a route between a source and destination <u>terminal</u>. The route can carry a variety of digital signals, such as triggers, clocks, and hardware events.

The source and destination terminals can be on different devices as long as a connecting public bus, such as RTSI or the PXI backplane, connects the devices. The DAQmx Connect Terminals VI does not modify a task. When the VI runs, the route is immediately reserved and committed to hardware. This type of routing is called <u>immediate routing</u>.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **Source terminal** specifies the originating <u>terminal</u> of the route. A <u>DAQmx terminal constant</u> lists all terminals available on devices installed in the system. You also can specify a source terminal by wiring a string that contains a <u>terminal name</u>.
- **destination terminal** specifies the receiving <u>terminal</u> of the route. A <u>DAQmx terminal constant</u> provides a list of all terminals available on devices installed in the system. You also can specify a destination terminal by wiring a string that contains a <u>terminal</u> <u>name</u>.
- **invert polarity** specifies whether to invert the signal this VI routes from the source terminal to the destination terminal. If the device is not capable of signal inversion or if a previous route reserved the inversion circuitry in an incompatible configuration, attempting to invert the signal causes an error.

no (0)	Do not invert the signal.
yes (1)	Invert the signal.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Disconnect Terminals

Removes signal routes you created by using the <u>DAQmx Connect</u> <u>Terminals</u> VI. The DAQmx Disconnect Terminals VI cannot remove taskbased routes, such as those you create through timing and triggering configuration.

When this VI runs, it immediately removes the route. This type of routing is called <u>immediate routing</u>.



- **Source terminal** specifies the originating <u>terminal</u> of the route. A <u>DAQmx terminal constant</u> lists all terminals available on devices installed in the system. You also can specify a source terminal by wiring a string that contains a <u>terminal name</u>.
- **destination terminal** specifies the receiving <u>terminal</u> of the route. A <u>DAQmx terminal constant</u> provides a list of all terminals available on devices installed in the system. You also can specify a destination terminal by wiring a string that contains a <u>terminal</u> <u>name</u>.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Tristate Output Terminal

Sets a terminal to <u>high-impedance state</u>. If you connect an external signal to a terminal on the I/O connector, the terminal must be in high-impedance state. Otherwise, the device could double-drive the terminal and damage the hardware. If you use this VI on a terminal in an active route, the VI fails and returns an error.

The <u>DAQmx Reset Device</u> VI sets all terminals on the I/O connector to high-impedance state but aborts any running tasks associated with the device.



- **output terminal** specifies the terminal on the I/O connector to set to high-impedance state. A <u>DAQmx terminal constant</u> lists all available terminals on installed devices. You also can specify an output terminal by using a string that contains a <u>terminal name</u>.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx System Setup VI and Functions

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx System Setup VI and functions to set up and query information about devices and software configuration.

Palette Object	Description
DAQmx Device Name Constant	Lists all devices installed in the system. Right-click the constant, and select I/O Name Filtering from the shortcut menu to <u>limit the devices</u> that the constant displays and to limit what you can enter in the constant. If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Device Property Node.
DAQmx Device Property Node	A Property Node with the <u>DAQmx Device</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Set Power Up States	Updates the states to which to set physical channels on a device when the device powers up or when you reset the device. A device stores power up states in non-volatile memory that you can write to only a limited number of times. Therefore, you should use this VI as infrequently as possible. This VI writes to the non-volatile memory only if a setting you request is different from the one currently stored. This VI writes power up states in the sequential order of the power up states array. Therefore, any physical channels with multiple entries in that array use the highest array index.
DAQmx System Property Node	A Property Node with the <u>DAQmx System</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

Subpalette	Description
DAQmx Storage	Use the DAQmx Storage VIs and functions to save

<u>VIs and</u>	and delete tasks, global channels, and custom scales
Functions	in MAX.

DAQmx Device Name Constant

Lists all devices installed in the system. Right-click the constant, and select **I/O Name Filtering** from the shortcut menu to <u>limit the devices</u> that the constant displays and to limit what you can enter in the constant. If you right-click this constant and create a Property Node, the Property Node is for the constant itself and is not a DAQmx Device Property Node.

DAQmx Device Property Node

A Property Node with the <u>DAQmx Device</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Set Power Up States

Updates the states to which to set physical channels on a device when the device powers up or when you reset the device. A device stores power up states in non-volatile memory that you can write to only a limited number of times. Therefore, you should use this VI as infrequently as possible. This VI writes to the non-volatile memory only if a setting you request is different from the one currently stored. This VI writes power up states in the sequential order of the power up states array. Therefore, any physical channels with multiple entries in that array use the highest array index.

The instances of this <u>polymorphic VI</u> specify whether to set power up states for analog or digital physical channels.

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

Select an instance 🔽

Digital State

Updates power up states for digital physical channels.



- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- **power up states** contains the physical channels and power up states to set. Each element of the array contains a physical channel and the power up state to set for that physical channel.
 - **physical channel** is the digital line or port to modify. You cannot modify dedicated digital input lines. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input. If you have an array of lines or ports, use the <u>DAQmx Flatten</u> <u>Channel String</u> VI to convert the array to a list.
 - **power up state** is the power up state to set for the physical channel specified with the **physical channel** input.

High (10192)	High logic.
Low (10214)	Low logic.
Tristate (10310)	High-impedance state. You can set this power up state only on devices with bidirectional ports, and you can set it only on entire ports. You cannot set this power up state on dedicated digital output lines.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **device out** is the name of the device the operation applied to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Logic Family

Sets the digital logic family to use when the device powers up.



- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- **Iogic family** specifies the logic family to set the device to when it powers up. A logic family corresponds to voltage thresholds that are compatible with a group of voltage standards. Refer to device documentation for information on the logic high and logic low voltages for these logic families.

2.5 V (14620)	Compatible with	CMOS signals.
----------------------	-----------------	---------------

3.3 V (14621) Compatible with LVTTL and LVCMOS signals.

5.0 V (14619) Compatible with TTL and CMOS signals.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

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

- **device out** is the name of the device the operation applied to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Analog

Updates power up states for analog physical channels.



- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- **voltage power up states** contains the physical channels and power up states to set. Each element of the array contains a physical channel and the power up state to set for that physical channel. You can set voltage power up states only for physical channels that support voltage output.
 - **physical channel** is the physical channel to modify. The <u>DAQmx physical channel constant</u> lists all physical channels for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the <u>DAQmx Flatten Channel String</u> VI to convert the array to a list.
 - **value** is the power up state to set for the physical channel specified with the **physical channel** input.
- **current power up states** contains the physical channels and power up states to set. Each element of the array contains a physical channel and the power up state to set for that physical channel. You can set current power up states only for physical channels that support current output.
 - **physical channel** is the physical channel to modify. The DAQmx physical channel constant lists all physical channels for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of physical channels to this input. If you have an array of physical channels, use the DAQmx Flatten Channel String VI to convert the array to a list.
 - **value** is the power up state to set for the physical channel

specified with the **physical channel** input.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **device out** is the name of the device the operation applied to.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the

error.

DAQmx Storage VIs and Functions

Owning Palette: DAQmx System Setup VI and Functions

Use the DAQmx Storage VIs and functions to save and delete tasks, global channels, and custom scales in MAX.

Palette Object	Description
DAQmx Delete Saved Task	Deletes the specified <u>task</u> from MAX. This VI does not clear the copy of the task stored in memory. Use the <u>DAQmx Clear</u> <u>Task</u> VI to clear that copy of the task.
DAQmx Persisted Channel Property Node	A Property Node with the <u>DAQmx Persisted Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Persisted Scale Property Node	A Property Node with the <u>DAQmx Persisted Scale</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Persisted Task Property Node	A Property Node with the <u>DAQmx Persisted Task</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Save</u> <u>Global</u> <u>Channel</u>	Saves the specified <u>local or global channel</u> to MAX as a global channel. You must specify both the local or global channel to save and a task that contains that channel.
<u>DAQmx</u> <u>Save</u> <u>Global</u>	Saves the specified <u>local or global channel</u> to MAX as a global channel. You must specify both the local or global channel to save and a task that contains that channel.

<u>Channel</u>	
<u>DAQmx</u> <u>Save</u> <u>Scale</u>	Saves the specified <u>custom scale</u> to MAX.
<u>DAQmx</u> <u>Save</u> <u>Scale</u>	Saves the specified <u>custom scale</u> to MAX.
<u>DAQmx</u> <u>Save</u> Task	Saves the specified <u>task</u> and any <u>local channels</u> it contains to MAX. This VI does not save global channels. Use the <u>DAQmx</u> <u>Save Global Channel</u> VI to save global channels.

DAQmx Delete Saved Global Channel

Deletes the specified <u>global channel</u> from MAX. This VI does not remove the global channel from <u>tasks</u> that use it.

channel name	DAQmx
error in	error out

- **channel name** is the global channel you want to delete.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

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

DAQmx Delete Saved Scale

Deletes the specified <u>custom scale</u> from MAX. This VI does not remove the custom scale from <u>virtual channels</u> that use it.

scale name ************************************	
error in and a error o	ut

- **scale name** is the custom scale to delete.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

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

DAQmx Delete Saved Task

Deletes the specified <u>task</u> from MAX. This VI does not clear the copy of the task stored in memory. Use the <u>DAQmx Clear Task</u> VI to clear that copy of the task.



- **task in** is the name of the task to which this operation applies.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function completes execution.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

DAQmx Persisted Channel Property Node

A Property Node with the <u>DAQmx Persisted Channel</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Persisted Scale Property Node

A Property Node with the <u>DAQmx Persisted Scale</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**
is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Persisted Task Property Node

A Property Node with the <u>DAQmx Persisted Task</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Save Global Channel

Saves the specified <u>local or global channel</u> to MAX as a global channel. You must specify both the local or global channel to save and a task that contains that channel.

Programmatically saved global channels cannot be viewed in NI-DAQ 7.3 or earlier. To view a programmatically saved global channel in an earlier version of NI-DAQ, first use MAX to open the global channel in NI-DAQ 7.4 or later and resave it.

Refer to the <u>DAQmx Professional Developer Tools</u> Web site for more information and examples of programmatically saving global channels.



- **task/channels in** is the task that contains the local or global channel you want to save. If you use a <u>DAQmx global channel</u> constant or control to select a global channel to modify and save, wire that constant or control to both this input and the **channel name** input.
- **channel name** is the local or global channel you want to save. The channel must be inside the task specified with the **task/channels in** input. If you use a <u>DAQmx global channel</u> constant or control to select a global channel to modify and save, wire that constant or control to both this input and the **task/channels in** input.
- **save as** is the name to save the task, global channel, or custom scale as. If you do not wire a value to this input, NI-DAQmx uses the name currently assigned to the task, global channel, or custom scale.
- **author** is a name to store with the task, global channel, or custom scale.
- **error in** describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before

this VI or function runs, the VI or function passes the **error in** value to **error out**. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- allow interactive editing? specifies whether to allow the task, global channel, or custom scale to be edited in the DAQ Assistant. If allow interactive editing? is TRUE, the DAQ Assistant must support all task or global channel settings.
- **allow interactive deletion?** specifies whether to allow the task, global channel, or custom scale to be deleted through MAX.
- **overwrite existing channel?** specifies whether to overwrite a global channel of the same name if one is already saved in MAX. If this input is FALSE and a global channel of the same name is already saved in MAX, this VI returns an error.
- **task out** is the same task or global channels you wired to the **task/channels in** input. This output is not the saved task or global channels.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

DAQmx Save Scale

Saves the specified custom scale to MAX.

Refer to the <u>DAQmx Professional Developer Tools</u> Web site for more information and examples of programmatically saving custom scales.



- **scale name** is the custom scale to save.
- **save as** is the name to save the task, global channel, or custom scale as. If you do not wire a value to this input, NI-DAQmx uses the name currently assigned to the task, global channel, or custom scale.
- **author** is a name to store with the task, global channel, or custom scale.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- allow interactive deletion? specifies whether to allow the task, global channel, or custom scale to be deleted through MAX.
- allow interactive editing? specifies whether to allow the task, global channel, or custom scale to be edited in the DAQ Assistant. If allow interactive editing? is TRUE, the DAQ Assistant must support all task or global channel settings.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Save Task

Saves the specified <u>task</u> and any <u>local channels</u> it contains to MAX. This VI does not save global channels. Use the <u>DAQmx Save Global Channel</u> VI to save global channels.

Programmatically saved tasks cannot be viewed in NI-DAQ 7.3 or earlier. To view a programmatically saved task in an earlier version of NI-DAQ, first use MAX to open the task in NI-DAQ 7.4 or later and resave it.

Refer to the <u>DAQmx Professional Developer Tools</u> Web site for more information and examples of programmatically saving tasks.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **save as** is the name to save the task, global channel, or custom scale as. If you do not wire a value to this input, NI-DAQmx uses the name currently assigned to the task, global channel, or custom scale.
- **author** is a name to store with the task, global channel, or custom scale.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

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

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- allow interactive editing? specifies whether to allow the task, global channel, or custom scale to be edited in the DAQ Assistant. If allow interactive editing? is TRUE, the DAQ Assistant must support all task or global channel settings.
- allow interactive deletion? specifies whether to allow the task, global channel, or custom scale to be deleted through MAX.
- **overwrite existing task?** specifies whether to overwrite a task of the same name if one is already saved in MAX. If this input is FALSE and a task of the same name is already saved in MAX, this VI returns an error.
- **task out** is the same task or global channels you wired to the **task/channels in** input. This output is not the saved task or global channels.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the

error.

DAQmx System Property Node

A Property Node with the <u>DAQmx System</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx TEDS VIs and Function

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx TEDS VIs and function to configure physical channels to use and to retrieve TEDS information from a physical channel.

Palette Object	Description	
<u>DAQmx</u> <u>Clear</u> TEDS	Removes TEDS information from the physical channel you specify. This VI temporarily overrides any TEDS configuration for the physical channel that you performed in MAX.	
DAQmx Configure TEDS	Associates TEDS information with the physical channel you specify. If you do not specify the filename of a data sheet in the virtual TEDS file input, this VI attempts to find a TEDS sensor connected to the physical channel. This VI temporarily overrides any TEDS configuration for the physical channel that you performed in MAX.	
DAQmx Physical Channel Property Node	A Property Node with the <u>DAQmx Physical Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.	
DAQmx Write TEDS Data	Writes <u>TEDS</u> data to the TEDS sensor connected to the physical channel you specify. This VI can write TEDS data from a 1D array of 8-bit unsigned integers or from a virtual TEDS file.	

DAQmx Clear TEDS

Removes TEDS information from the physical channel you specify. This VI temporarily overrides any TEDS configuration for the physical channel that you performed in MAX.



- **physical channel** is the name of the physical channel you want to configure. A <u>DAQmx physical channel constant</u> or control lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>physical channel name</u> to this input.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out

indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Configure TEDS

Associates TEDS information with the physical channel you specify. If you do not specify the filename of a data sheet in the **virtual TEDS file** input, this VI attempts to find a TEDS sensor connected to the physical channel. This VI temporarily overrides any TEDS configuration for the physical channel that you performed in MAX.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **physical channel** is the name of the physical channel you want to configure. A <u>DAQmx physical channel constant</u> or control lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>physical channel name</u> to this input.
- virtual TEDS file is the path to a Virtual TEDS data sheet that you want to associate with the physical channel. If you do not wire anything to this input, this VI attempts to find a TEDS sensor connected to the physical channel.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Physical Channel Property Node

A Property Node with the <u>DAQmx Physical Channel</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Write TEDS Data

Writes <u>TEDS</u> data to the TEDS sensor connected to the **physical channel** you specify. This VI can write TEDS data from a 1D array of 8bit unsigned integers or from a virtual TEDS file.

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

Select an instance 🗾

U8 Array

Writes data from a 1D array of 8-bit unsigned integers to the TEDS sensor.



- **physical channel** is the name of the physical channel you want to configure. A <u>DAQmx physical channel constant</u> or control lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>physical channel name</u> to this input.
- **TEDS bitstream** is the TEDS bitstream to write to the sensor. This bitstream must be constructed according to the IEEE 1451.4 specification.

National Instruments provides a LabVIEW library for viewing and editing TEDS bitstreams and virtual TEDS files. You can download the TEDS Library for LabVIEW at <u>ni.com/pnp</u>.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

basic TEDS data options specifies how to handle <u>basic TEDS</u> <u>data</u> in the bitstream.

Do Not Write (12540)	Ignore basic TEDS data.
Write To EEPROM (12538)	Write basic TEDS data to the EEPROM, even if the sensor includes a PROM. You cannot write basic TEDS data if the PROM contains data.
Write To PROM Once (12539)	Write basic TEDS data to the PROM. Any subsequent attempts to write basic TEDS data result in an error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Virtual TEDS File

Writes data from a virtual TEDS file to the TEDS sensor.



- **physical channel** is the name of the physical channel you want to configure. A <u>DAQmx physical channel constant</u> or control lists all physical channels on devices and modules installed in the system. You also can wire a string that contains a <u>physical channel name</u> to this input.
- virtual TEDS file specifies the filename of a virtual TEDS file that contains the bitstream to write.

National Instruments provides a LabVIEW library for viewing and editing TEDS bitstreams and virtual TEDS files. You can download the TEDS Library for LabVIEW at <u>ni.com/pnp</u>.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

basic TEDS data options specifies how to handle <u>basic TEDS</u> <u>data</u> in the bitstream.

Do Not Write (12540)	Ignore basic TEDS data.
Write To EEPROM (12538)	Write basic TEDS data to the EEPROM, even if the sensor includes a PROM. You cannot write basic TEDS data if the PROM contains data.
Write To PROM Once (12539)	Write basic TEDS data to the PROM. Any subsequent attempts to write basic TEDS data result in an error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Utilities VIs

Owning Palette: DAQmx Advanced VIs and Functions

Use the DAQmx Utilities VIs as helper VIs to make NI-DAQmx programming easier.

Palette Object	Description
<u>DAQmx</u> <u>Flatten</u> <u>Channel</u> <u>String</u>	Converts an array of physical or virtual channel names to a comma-delimited list of names. You can use this VI to convert an array of channel names to a single string prior to using the DAQmx Create Virtual Channel VI or the DAQmx Create Task VI.
DAQmx Unflatten Channel String	Converts a comma-delimited list or range of physical or virtual channels into an array of physical or virtual channel names.

DAQmx Flatten Channel String

Converts an array of physical or virtual channel names to a commadelimited list of names. You can use this VI to convert an array of channel names to a single string prior to using the <u>DAQmx Create Virtual Channel</u> VI or the <u>DAQmx Create Task</u> VI.



- **names in** is the array of physical or virtual channel names.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **names out** is the resulting comma-delimited list of physical or virtual channel names.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out

indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Unflatten Channel String

Converts a comma-delimited list or range of physical or virtual channels into an array of physical or virtual channel names.



- **names in** is the list or range of physical or virtual channels.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **names out** is the array of physical or virtual channel names. Each element of the array contains a single channel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

DAQmx Advanced Task Options VIs and Functions

Owning Palette: DAQmx - Data Acquisition VIs and Functions

Use the DAQmx Advanced Task Options VIs and functions for advanced configuration and control of <u>tasks</u>.

Palette Object	Description
<u>DAQmx</u> <u>Buffer</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Buffer</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Configure Input Buffer	Overrides the <u>automatic input buffer allocation</u> that NI- DAQmx performs.
DAQmx Configure Output Buffer	Overrides the <u>automatic output buffer allocation</u> that NI- DAQmx performs.
<u>DAQmx</u> <u>Control</u> Task	Alters the <u>state</u> of a task according to the action you specify.
<u>DAQmx</u> <u>Create</u> <u>Task</u>	Creates a <u>task</u> and adds <u>virtual channels</u> to that task if you specify them in the global virtual channels input. If you specify a task to copy , this VI duplicates the configuration of the specified task in the newly created task before it adds any additional global virtual channels.
<u>DAQmx</u> <u>Is Task</u> Done	Queries the status of the task and indicates if it completed execution. Use this VI to <u>ensure that the specified operation is complete</u> before you stop the task.
DAQmx Send Software	Generates the software trigger you specify. You must configure the software trigger with the <u>DAQmx Trigger</u> <u>Property Node</u> before you use this VI.

<u>Trigger</u>	
DAQmx	A Property Node with the <u>DAQmx Task</u> class preselected.
<u>Task</u>	Right-click the Property Node and choose Select Filter from
Property	the shortcut menu to make the Property Node show only the
Node	properties supported by a particular device installed in the
	system or supported by all the devices installed in the system.

Subpalette	Description
DAQmx Export Signals VI and Functions	Use the DAQmx Export Signals VI and functions to configure and route control signals within a device or from one device to another.

DAQmx Buffer Property Node

A Property Node with the <u>DAQmx Buffer</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Configure Input Buffer

Overrides the automatic input buffer allocation that NI-DAQmx performs.

task/channels in CARMAN task out buffer size (in samples per... error out

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **buffer size (in samples per channel)** is the number of samples the buffer can hold for each channel in the task. Zero indicates to allocate no buffer. Use a buffer size of 0 to perform a hardwaretimed operation without using a buffer.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
DAQmx Configure Output Buffer

Overrides the <u>automatic output buffer allocation</u> that NI-DAQmx performs.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **buffer size (in samples per channel)** is the number of samples the buffer can hold for each channel in the task. Zero indicates to allocate no buffer. Use a buffer size of 0 to perform a hardwaretimed operation without using a buffer.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-

DAQmx creates this task automatically.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Control Task

Alters the state of a task according to the action you specify.

If **error in** indicates that an error occurred previously, this VI executes normally if **action** is **unreserve** or **abort**.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **action** specifies how to alter the task state.

abort (6)	Aborts execution of the task. Aborting a task immediately terminates the currently active operation, such as a read or a write. Aborting a task puts the task into an unstable but recoverable state. To recover the task, use DAQmx Start to restart the task or use DAQmx Stop to reset the task without starting it.
commit (3)	Programs the hardware as much as possible according to the task configuration.
reserve (4)	Reserves the hardware resources needed for the task. No other tasks can reserve these same resources.
unreserve (5)	Releases all previously reserved resources.
verify (2)	Verifies that all task parameters are valid for the hardware.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Create Task

Creates a <u>task</u> and adds <u>virtual channels</u> to that task if you specify them in the **global virtual channels** input. If you specify a **task to copy**, this VI duplicates the configuration of the specified task in the newly created task before it adds any additional global virtual channels.

If you use this VI within a loop, NI-DAQmx creates a new task in each iteration of the loop. Use the <u>DAQmx Clear Task</u> VI within the loop after you are finished with the task to avoid allocating unnecessary memory. Refer to <u>Task Creation and Destruction</u> for more information on when NI-DAQmx creates tasks and when LabVIEW automatically destroys tasks.



- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task to copy** is the name of a task to make a copy of.
- **global virtual channels** specifies a global virtual channel or list of global virtual channels to add to the task.

A <u>DAQmx global channel constant</u> lists all global virtual channels on the system.

If you have an array of global virtual channels, use the <u>DAQmx</u> <u>Flatten Channel String</u> VI to convert the array to a list.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

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

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **new task name** is the name to assign to the task. If you use this VI in a loop and specify a name for the task, you must use the <u>DAQmx Clear Task</u> VI within the loop after you are finished with the task. Otherwise, NI-DAQmx attempts to create multiple tasks with the same name, which results in an error. Refer to <u>Task</u> <u>Creation and Destruction</u> for more information.
- **auto cleanup** specifies if LabVIEW automatically destroys the task when the application completes execution. If **auto cleanup** is FALSE, LabVIEW does not automatically destroy the task until you exit LabVIEW. Use the <u>DAQmx Clear Task</u> VI to manually destroy the task.
- **task out** is a reference to the new task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Export Signals VI and Functions

Owning Palette: <u>DAQmx Advanced Task Options VIs and Functions</u> Use the DAQmx Export Signals VI and functions to configure and route control signals within a device or from one device to another.

Palette Object	Description
DAQmx Export Signal Property Node	A Property Node with the <u>DAQmx Export Signal</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Export</u> <u>Signal</u>	Routes a <u>control signal</u> to the <u>terminal</u> you specify. The output terminal can reside on the device that generates the control signal or on a different device. You can use this VI to share clocks and triggers among multiple tasks and devices. The routes this VI creates are <u>task-based</u> routes.
<u>DAQmx</u> Terminal Constant	Lists all <u>terminals</u> available on devices installed in the system. Select Browse to select multiple terminals. Right-click the constant and select I/O Name Filtering from the shortcut menu to <u>limit the terminals</u> that the constant displays and to limit what you can enter in the constant.

DAQmx Export Signal

Routes a <u>control signal</u> to the <u>terminal</u> you specify. The output terminal can reside on the device that generates the control signal or on a different device. You can use this VI to share clocks and triggers among multiple tasks and devices. The routes this VI creates are <u>task-based</u> routes.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **signal** is the name of the trigger, clock, or event to export.

10 MHz Reference Clock (12536)	Output of an oscillator that you can use to synchronize multiple devices.
20Mhz Timebase Clock (12486)	Output of an oscillator that is the onboard source of the Master Timebase. Other timebases are derived from this clock.
Advance Complete Event (12492)	Signal a switch product generates after it both executes the command(s) in a scan list entry and waits for the settling time to elapse.
Advance Trigger (12488)	Trigger that moves a switch to the next entry in a scan list.
AI Convert Clock (12484)	Clock that causes an analog-to-digital conversion on a <u>multiplexed</u> device. One conversion corresponds to a single sample from one channel.

Al Hold Complete Event (12493)	Signal a <u>multiplexed</u> device generates when the device latches analog input data (the ADC enters "hold" mode) and it is safe for any external switching hardware to remove the signal and replace it with the next signal. This event does not indicate the completion of the actual analog-to-digital conversion.
Change Detection Event (12511)	Signal a static DIO device generates when the device detects a rising or falling edge on any of the lines or ports you selected when you configured change detection timing.
Counter Output Event (12494)	Signal a counter generates. Each time the counter reaches terminal count, this signal toggles or pulses.
Reference Trigger (12490)	Trigger that establishes the reference point between pretrigger and posttrigger samples.
Sample Clock (12487)	Clock the device uses to time each sample.
Start Trigger (12491)	Trigger that begins a measurement or generation.
Watchdog Timer Expired Event (12512)	Signal a static DIO device generates when the watchdog timer expires.

- **Output terminal** is the destination of the exported signal. A <u>DAQmx terminal constant</u> lists all terminals on installed devices.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Export Signal Property Node

A Property Node with the <u>DAQmx Export Signal</u> class preselected. Rightclick the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Terminal Constant

Lists all <u>terminals</u> available on devices installed in the system. Select **Browse** to select multiple terminals. Right-click the constant and select **I/O Name Filtering** from the shortcut menu to <u>limit the terminals</u> that the constant displays and to limit what you can enter in the constant.

If you right-click this constant and create a Property Node, the Property Node is for the constant itself and does not provide any configuration options for terminals.

DAQmx Is Task Done

Queries the status of the task and indicates if it completed execution. Use this VI to <u>ensure that the specified operation is complete</u> before you stop the task.

task/channels in ~~~	DAQmx task out
error in 🚥	? task done?

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **task done?** indicates if the measurement or generation completed.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Send Software Trigger

Generates the software trigger you specify. You must configure the software trigger with the <u>DAQmx Trigger Property Node</u> before you use this VI.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **trigger** specifies the software trigger to generate.

Advance Trigger (12488) Generate the advance trigger.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-

DAQmx creates this task automatically.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Device Configuration VIs and Functions

Owning Palette: DAQmx - Data Acquisition VIs and Functions

Use the DAQmx Device Configuration VIs and functions for hardwarespecific configuration and control.

Palette Object	Description
DAQmx Reset Device	Immediately aborts all active tasks associated with a device, disconnects any routes, and returns the device to an <u>initialized</u> state. Aborting a task immediately terminates the currently active operation, such as a read or a write. Aborting a task puts the task into an unstable but recoverable state. To recover the task, use DAQmx Start to restart the task or use DAQmx Stop to reset the task without starting it.

Subpalette	Description
DAQmx Switches VIs and Functions	Use the DAQmx Switches VIs and functions to configure and control NI switch products.
DAQmx Watchdog VIs and Function	Use the DAQmx Watchdog VIs and function to configure and control the watchdog timer of a device.

DAQmx Reset Device

Immediately aborts all active tasks associated with a device, disconnects any routes, and returns the device to an <u>initialized</u> state. Aborting a task immediately terminates the currently active operation, such as a read or a write. Aborting a task puts the task into an unstable but recoverable state. To recover the task, use DAQmx Start to restart the task or use DAQmx Stop to reset the task without starting it.

Using this VI on a switch device resets the topology of that device to the default switch topology configured in MAX.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switches VIs and Functions

Owning Palette: <u>DAQmx Device Configuration VIs and Functions</u> Use the DAQmx Switches VIs and functions to configure and control NI switch products.

Palette Object	Description
<u>DAQmx</u> <u>Switch</u> <u>Channel</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Switch Channel</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
<u>DAQmx</u> <u>Switch</u> <u>Connect</u>	Makes immediate connections between switch channels. The DAQmx Switch Connect VI attempts to find the shortest available path between the channels. If a path is not available, this VI returns an error. The instances of this <u>polymorphic VI</u> make a single connection between two switch channels or make connections between multiple switch channels.
<u>DAQmx</u> <u>Switch</u> <u>Create</u> <u>Scan List</u>	Creates a new switch scanning task with the sequence you specify in the scan list input.
<u>DAQmx</u> <u>Switch</u> <u>Device</u> <u>Property</u> <u>Node</u>	A Property Node with the <u>DAQmx Switch Device</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Switch Disconnect All	Terminates all active connections on the device, which places the relays into the topology reset state. You can use this VI to terminate connections you made by using the <u>DAQmx Switch Connect</u> VI, the <u>DAQmx Switch Open</u> <u>Relays</u> VI, or the <u>DAQmx Switch Close Relays</u> VI. This VI does not alter the settling time or channel usage settings.

	Use the <u>DAQmx Reset Device</u> VI or the <u>DAQmx Switch Set</u> <u>Topology and Reset</u> VI to reset those settings.
DAQmx Switch Disconnect	Immediately terminates switch connections you created by using the <u>DAQmx Switch Connect</u> VI. This VI returns an error if you attempt to terminate a connection that does not exist. The instances of this <u>polymorphic</u> VI terminate a single connection or multiple connections.
<u>DAQmx</u> <u>Switch</u> <u>Find Path</u>	Returns information about the path between switch channel 1 and switch channel 2 . If the channels are connected, this VI returns the connection path. If the channels are not connected, the VI returns the shortest available connection path if a path is available.
DAQmx <u>Switch</u> <u>Scan</u> Property <u>Node</u>	A Property Node with the <u>DAQmx Switch Scan</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Switch Set Topology and Reset	Resets a switch device and sets its topology to the one you specify in topology name . Use this VI to initialize a switch device before scanning or making connections to ensure the initial state of the switch device.
DAQmx Switch Wait for Settling	Waits for the settling time on the device to expire. The device resets this time and begins counting down each time the device performs an operation. Therefore, this VI could return immediately if no operation happened recently.

Subpalette	Description
DAQmx Relay Operations VIs	Use the DAQmx Relay Operations VIs to configure and query switch relays.

DAQmx Relay Operations VIs

Owning Palette: DAQmx Switches VIs and Functions

Use the DAQmx Relay Operations VIs to configure and query switch relays.

Palette Object	Description	
DAQmx Switch Close Relays	Closes the relays you specify. If you set wait for settling to TRUE, this VI waits only after closing all relays. If you want to wait for settling after closing each relay, use this VI multiple times to close each relay separately.	
DAQmx Switch Get Relay Count	Returns the number of times a relay has actuated on switches that support querying the relay count. You must wait for the switch to settle before this VI can determine an accurate relay count. Use this VI to track relay lifetime and usage. The instances of this <u>polymorphic</u> VI can query the count of a single relay or multiple relays.	
DAQmx Switch Get Relay Position	Returns the current position of the relay or relays you specify. The instances of this <u>polymorphic</u> VI return the position of a single relay or multiple relays.	
DAQmx <u>Switch</u> Open Relays	Opens the relays you specify. If you set wait for settling to TRUE, this VI waits only after opening all relays. If you want to wait for settling after opening each relay, use this VI multiple times to open each relay separately.	

DAQmx Switch Close Relays

Closes the relays you specify. If you set **wait for settling** to TRUE, this VI waits only after closing all relays. If you want to wait for settling after closing each relay, use this VI multiple times to close each relay separately.

When you operate relays directly, you circumvent the protection channel <u>usage types</u> offer. Avoid using this VI when you use the <u>DAQmx Switch</u> <u>Connect</u> VI or the <u>DAQmx Switch Disconnect</u> VI. This VI does not pass relay changes to those VIs.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

relays is the relay or set of relays to close. A <u>DAQmx switch</u> <u>constant</u> lists all relays available on devices installed in the system. Right-click the constant, and select **Browse** from the shortcut menu to select multiple relays. You also can supply a string that contains a comma-delimited list of relays. Refer to the switch module specifications for relay names and locations.

If you have an array of relays, use the <u>DAQmx Flatten Channel</u> <u>String</u> VI to convert the array to a list.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

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

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Get Relay Count

Returns the number of times a relay has actuated on switches that support querying the relay count. You must wait for the switch to settle before this VI can determine an accurate relay count. Use this VI to track relay lifetime and usage. The instances of this <u>polymorphic</u> VI can query the count of a single relay or multiple relays.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Single

Returns the number of times a single relay has actuated.



relay is the relay to query. A <u>DAQmx switch constant</u> lists all relays on switch devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **count**is the number of times the relay has actuated.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

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

Multiple

Returns the number of times a set of relays have actuated.



- **relays** is the set of relays to query. A <u>DAQmx switch constant</u> lists all relays available on devices installed in the system. Right-click the constant, and select **Browse** from the shortcut menu to select multiple relays. You also can supply a string that contains a comma-delimited list of relays.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **counts** is number of times each relay has actuated. The order of this array corresponds to the order of **relays**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

DAQmx Switch Get Relay Position

Returns the current position of the relay or relays you specify. The instances of this <u>polymorphic</u> VI return the position of a single relay or multiple relays.

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

Select an instance 🔽

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Single

Returns the current position of a single relay.



relay is the relay to query. A <u>DAQmx switch constant</u> lists all relays on switch devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **position** is the position of the relay.

closed (10438)	Relay is closed.
open (10437)	Relay is open.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the

shortcut menu for more information about the error.

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

Multiple

Returns the current position of a set of relays.



- **relays** is the set of relays to query. A <u>DAQmx switch constant</u> lists all relays available on devices installed in the system. Right-click the constant, and select **Browse** from the shortcut menu to select multiple relays. You also can supply a string that contains a comma-delimited list of relays.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **positions** is the position of each relay. The order of this array corresponds to the order of **relays**.

closed (10438)	Relay is closed.
open (10437)	Relay is open.

error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, **error out** describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Switch Open Relays

Opens the relays you specify. If you set **wait for settling** to TRUE, this VI waits only after opening all relays. If you want to wait for settling after opening each relay, use this VI multiple times to open each relay separately.

When you operate relays directly, you circumvent the protection offered by channel <u>usage types</u>. Avoid using this VI when you use the <u>DAQmx</u> <u>Switch Connect</u> VI or the <u>DAQmx Switch Disconnect</u> VI. This VI does not pass relay changes to those VIs.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

relays is the relay or set of relays to open. A <u>DAQmx switch</u> <u>constant</u> lists all relays available on devices installed in the system. Right-click the constant, and select **Browse** from the shortcut menu to select multiple relays. You also can supply a string that contains a comma-delimited list of relays. Refer to the switch module specifications for relay names and locations.

If you have an array of relays, use the <u>DAQmx Flatten Channel</u> <u>String</u> VI to convert the array to a list.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
code is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Channel Property Node

A Property Node with the <u>DAQmx Switch Channel</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Switch Connect

Makes immediate connections between switch channels. The DAQmx Switch Connect VI attempts to find the shortest available path between the channels. If a path is not available, this VI returns an error. The instances of this <u>polymorphic VI</u> make a single connection between two switch channels or make connections between multiple switch channels.

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

Select an instance 🗾

Single

Makes a connection between switch channel 1 and switch channel 2.



- **switch channel 1** is the first channel in the connection. A <u>DAQmx</u> <u>switch name constant</u> lists all switch channels available on devices installed in the system.
- **switch channel 2** is the second channel in the connection. A <u>DAQmx switch name constant</u> lists all switch channels available on devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Multiple

Makes the connections you specify in **connection list**. You can specify the two endpoints only or the explicit path between the two endpoints. This VI can make connections on multiple devices, but each individual connection must reside on a single device. In the event of an error, connecting stops at the point in the list where the error occurred. If you set **wait for settling** to TRUE, this VI waits only after making all connections. If you want to wait for settling between connections, use this VI multiple times to make connections separately.



connection list uses a special <u>syntax</u> to list the connections to make between switch channels.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns

immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Create Scan List

Creates a new switch scanning task with the sequence you specify in the **scan list** input.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **scan list** uses a special <u>syntax</u> to specify the sequence of connections and disconnections for the task.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task name** is the name to assign to the task this VI creates.

auto cleanup specifies if LabVIEW automatically destroys the task when the application completes execution. If **auto cleanup** is FALSE, LabVIEW does not automatically destroy the task until you exit LabVIEW. Use the <u>DAQmx Clear Task</u> VI to manually destroy the task.

- **task out** is a reference to the task after this VI or function completes execution.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Device Property Node

A Property Node with the <u>DAQmx Switch Device</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Switch Disconnect

Immediately terminates switch connections you created by using the <u>DAQmx Switch Connect</u> VI. This VI returns an error if you attempt to terminate a connection that does not exist. The instances of this <u>polymorphic</u> VI terminate a single connection or multiple connections.

Using this VI on topologies with multiplexers or form C relays does not necessarily imply an electrical disconnect because disconnecting channels in these topologies might connect another channel as a side effect. The DAQmx Switch Disconnect VI frees such channels for use in other connections.

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

Select an instance 🔽

Single

Terminates a connection you specify with the endpoints **switch channel 1** and **switch channel 2**.



- **switch channel 1** is the first channel in the connection. A <u>DAQmx</u> <u>switch name constant</u> lists all switch channels available on devices installed in the system.
- **switch channel 2** is the second channel in the connection. A <u>DAQmx switch name constant</u> lists all switch channels available on devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Multiple

Terminates the connections you specify in **disconnection list**. This VI can terminate connections on multiple devices. In the event of an error, the VI stops at the point in the list where the error occurred. If you set **wait for settling** to TRUE, this VI waits only after terminating all connections. If you want to wait for settling between disconnections, use this VI multiple times to terminate connections separately.



disconnection list uses a special <u>syntax</u> to specify the list of switch connections to terminate.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Disconnect All

Terminates all active connections on the device, which places the relays into the topology reset state. You can use this VI to terminate connections you made by using the <u>DAQmx Switch Connect</u> VI, the <u>DAQmx Switch Open Relays</u> VI, or the <u>DAQmx Switch Close Relays</u> VI. This VI does not alter the settling time or channel usage settings. Use the <u>DAQmx Reset Device</u> VI or the <u>DAQmx Switch Set Topology and Reset</u> VI to reset those settings.

If you set **wait for settling** to TRUE, this VI waits only after terminating all connections. If you want to wait for settling between disconnections, use the <u>DAQmx Switch Disconnect</u> VI multiple times to terminate connections separately.



- **switch device** is the name of the switch device this operation applies to. A <u>DAQmx device name constant</u> lists all devices, including switches, installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- wait for settling specifies whether this VI waits for the switches to settle before returning. If wait for settling is FALSE, the VI returns immediately after the operation. Use the <u>Settling Time</u> property to control the settling time.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Find Path

Returns information about the path between **switch channel 1** and **switch channel 2**. If the channels are connected, this VI returns the connection path. If the channels are not connected, the VI returns the shortest available connection path if a path is available.



- **switch channel 1** is the first channel in the connection. A <u>DAQmx</u> <u>switch name constant</u> lists all switch channels available on devices installed in the system.
- **switch channel 2** is the second channel in the connection. A <u>DAQmx switch name constant</u> lists all switch channels available on devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **path** is the existing path or an available path between **switch**

channel 1 and **switch channel 2**. This path uses the same <u>syntax</u> as a connection or disconnection list, and you can wire it directly to the <u>DAQmx Switch Connect</u> VI or the <u>DAQmx Switch Disconnect</u> VI.

path status is the status of the requested path.

channel in use (10434)	Path between the two endpoints is not available because another connection is already using a channel needed for routing.
channel reserved for routing (10436)	One of the endpoint channels is reserved for routing.
path already exists (10432)	The channels are already connected by path .
path available (10431)	path returns the available path between the channels.
path unsupported (10433)	No path is available between the channels. You might need to reserve more channels for routing in order to create an available path.
source channel conflict (10435)	No path is available between the two channels because connecting the channels would directly or indirectly connect two source channels.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

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

warning code.

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

DAQmx Switch Scan Property Node

A Property Node with the <u>DAQmx Switch Scan</u> class preselected. Rightclick the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- **task in** is the name of the task to which this operation applies.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function completes execution.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Switch Set Topology and Reset

Resets a switch device and sets its topology to the one you specify in **topology name**. Use this VI to initialize a switch device before scanning or making connections to ensure the initial state of the switch device.

This VI differs from the <u>DAQmx Reset Device</u> VI because it uses the input topology name rather than the one you specify in MAX. This VI does not modify the channel usage defaults you configure in MAX.



- **switch device** is the name of the switch device this operation applies to. A <u>DAQmx device name constant</u> lists all devices, including switches, installed in the system.
- **topology name** is the switch topology to use on the device. A <u>DAQmx switch constant</u> or control lists all topologies supported by devices installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function runs normally and passes the error in value to error out. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error,

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

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Switch Wait for Settling

Waits for the settling time on the device to expire. The device resets this time and begins counting down each time the device performs an operation. Therefore, this VI could return immediately if no operation happened recently.



- **switch device** is the name of the switch device this operation applies to. A <u>DAQmx device name constant</u> lists all devices, including switches, installed in the system.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out

indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

DAQmx Watchdog VIs and Function

Owning Palette: DAQmx Device Configuration VIs and Functions

Use the DAQmx Watchdog VIs and function to configure and control the watchdog timer of a device.

Palette Object	Description
DAQmx	Controls the watchdog timer task according to the action you
Control	specify. This VI does not program the watchdog timer on a
Watchdog	real-time controller. Use the Real-Time Watchdog VIs to
Task	program the watchdog timer on a real-time controller.
DAQmx	Creates and configures a task that controls the watchdog
Create	timer of a device. The timer activates when you start the task.
Watchdog	This VI does not program the watchdog timer on a real-time
Timer	controller. Use the Real-Time Watchdog VIs to program the
Task	watchdog timer on a real-time controller.
<u>DAQmx</u> Watchdog Property Node	A Property Node with the <u>DAQmx Watchdog</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all devices installed in the system.

DAQmx Control Watchdog Task

Controls the watchdog timer task according to the action you specify. This VI does not program the watchdog timer on a real-time controller. Use the Real-Time Watchdog VIs to program the watchdog timer on a real-time controller.

task in	DAOmx task out
error in	error out

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

task in is the name of the task to which this operation applies.

action specifies how to control the watchdog timer task.

clear expiration (1)	Unlock a device whose watchdog timer expired.
reset timer (0)	Reset the internal timer. You must continually reset the internal timer to prevent it from timing out and locking the device.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source

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

- **task out** is a reference to the task after this VI or function completes execution.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Create Watchdog Timer Task

Creates and configures a task that controls the watchdog timer of a device. The timer activates when you start the task. This VI does not program the watchdog timer on a real-time controller. Use the Real-Time Watchdog VIs to program the watchdog timer on a real-time controller.

The <u>DAQmx Watchdog Timer</u> properties contain additional watchdog timer configuration options.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **device in** is the name as configured in MAX of the device to which this operation applies. A <u>DAQmx device name constant</u> lists all devices installed in the system.
- timeout is the amount of time in seconds until the watchdog timer expires. A value of -1 means the internal timer never expires. Set this input to -1 if you use an Expiration Trigger to expire the watchdog task. If this time elapses, the device sets the physical channels to the states you specify with the digital physical channel expiration states input.

Use the <u>DAQmx Control Watchdog Task</u> VI with the **action** input set to **reset timer** to prevent the watchdog timer from expiring.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

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

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **new task name** is the name to assign to the task. If you use this VI in a loop and specify a name for the task, you must use the <u>DAQmx Clear Task</u> VI within the loop after you are finished with the task. Otherwise, NI-DAQmx attempts to create multiple tasks with the same name, which results in an error. Refer to <u>Task</u> <u>Creation and Destruction</u> for more information.
- auto cleanup specifies if LabVIEW automatically destroys the task when the application completes execution. If **auto cleanup** is FALSE, LabVIEW does not automatically destroy the task until you exit LabVIEW. Use the <u>DAQmx Clear Task</u> VI to manually destroy the task.
- digital physical channel expiration states contains the states to which to set digital physical channels when the watchdog timer expires. Each element of the array contains a digital physical channel name and the corresponding state for that digital physical channel.
 - **physical channel** is the digital line or port to modify. You cannot modify dedicated digital input lines. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in the system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input. If you have an array of lines or ports, use the <u>DAQmx Flatten</u> <u>Channel String</u> VI to convert the array to a list.
 - **expiration state** specifies the state to which to set the digital physical channel when the watchdog timer expires.

```
High High logic. (10192)
```

Low (10214)	Low logic.
No Change (10160)	Expiration does not affect the port. Do not change the state of any lines in the port, and do not lock the port. For example, if a line is high when the timer expires, that line stays high, and you can write new values to the line. You can select this value only for entire ports.
Tristate (10310)	High-impedance state. You can select this state only on devices with bidirectional ports, and you can select it only for entire ports. You cannot select this state for dedicated digital output lines.

- **task out** is a reference to the new task.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Watchdog Property Node

A Property Node with the <u>DAQmx Watchdog</u> class preselected. Rightclick the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all devices installed in the system.

- **task** is the name of the task to which this operation applies.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task** is a reference to the task after this function completes execution.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).
DAQmx Real Time VIs and Function

Owning Palette: <u>DAQmx - Data Acquisition VIs and Functions</u> Use the DAQmx Real Time VIs and function to configure and perform real-time operations.

Palette Object	Description
DAQmx Real- Time Property Node	A Property Node with the <u>DAQmx Real-Time</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.
DAQmx Wait For <u>Next</u> Sample <u>Clock</u>	Waits until the next pulse of the <u>Sample Clock</u> occurs. If an extra Sample Clock pulse occurs between calls to this VI, the second call returns an error or warning and waits for the next Sample Clock pulse. Use the <u>Convert Late Errors to Warnings</u> DAQmx Real-Time property to specify whether this VI returns errors or warnings. If that property is TRUE, any warnings this VI returns do not include the source string.

DAQmx Real-Time Property Node

A Property Node with the <u>DAQmx Real-Time</u> class preselected. Rightclick the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

- \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.
- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error

status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1..n** is an example of a property you want to get (read) or set (write).

DAQmx Timed Loop VI and Function (Not Available in LabVIEW 7.0)

Owning Palette: DAQmx Real Time VIs and Function

Use the DAQmx Timed Loop VI and function to create and configure timing sources for Timed Loops.

Palette Object	Description
DAQmx Create Timing Source	Creates a timing source for a Timed Loop. The timing source is based on signals a device generates. The instances of this polymorphic VI correspond to the type of signal to use for the timing source. Wire the timing source out output to the source input of the Input Node of a Timed Loop to use this timing source with that Timed Loop.
DAQmx Timing Source Property Node	A Property Node with the <u>DAQmx Timing Source</u> class preselected. Right-click the Property Node and choose Select Filter from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

DAQmx Create Timing Source (Not Available in LabVIEW 7.0)

Creates a timing source for a Timed Loop. The timing source is based on signals a device generates. The instances of this polymorphic VI correspond to the type of signal to use for the timing source. Wire the **timing source out** output to the **source** input of the Input Node of a Timed Loop to use this timing source with that Timed Loop.

Timing sources send ticks to a timed loop. Those ticks determine when the Timed Loop executes. By default, the Timed Loop executes each time it receives a tick from a timing source, but other Timed Loop settings can affect when it executes. For example, if you set **period** in the Loop Configuration dialog to 2, the Timed Loop executes on every other tick the timing source sends.

Refer to <u>NI-DAQmx 7.4 and Later Single-Point Real-Time Applications</u> for an overview of real-time architectures and applications.

The <u>DAQmx Timing Source</u> properties include additional timing source configuration options.

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

Select an instance

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

Control Loop From Task

Creates a timing source that uses a combination of the sample clock and the specified sleep time to determine when to send ticks to a Timed Loop. You must create an analog input task that uses sample clock timing and wire that task to the **task/channels in** input of this VI. Do not use the <u>DAQmx Start Task VI</u> to start the task. The Timed Loop starts the task automatically. National Instruments recommends that you use this instance instead of the <u>Signal From Task</u> instance for best performance with Timed Loops.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **sleep time (us)** specifies in microseconds the amount of time the Timed Loop sleeps after each sample clock pulse. The default sleep time is 0. This input is ignored on all OSes other than Pharlap.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

abc

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

- timing source name is the name to assign to the timing source this VI creates.
- **task out** is a reference to the task after this VI or function completes execution.
- **timing source out** is the name of the timing source this VI creates. Wire this output to the **source** input of the Input Node of a Timed Loop. If you specified a name for the timing source in the **timing source name** input, this output contains the same name. If you did not specify a name for the timing source, NI-DAQmx creates a name and sets this output to that name.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Frequency

Creates a timing source that sends ticks to a Timed Loop at a constant frequency. NI-DAQmx uses the **counter** you specify to generate the signal for the timing source.



- **counter** specifies the name of the counter to use for the timing source. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system. You also can wire a string that contains a <u>name</u> of a counter to this input.
- **frequency** specifies in hertz the frequency at which you want to send ticks to the Timed Loop.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **timing source name** is the name to assign to the timing source

this VI creates.

- timing source out is the name of the timing source this VI creates. Wire this output to the source input of the Input Node of a Timed Loop. If you specified a name for the timing source in the timing source name input, this output contains the same name. If you did not specify a name for the timing source, NI-DAQmx creates a name and sets this output to that name.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Digital Edge (Counter)

Creates a timing source that sends ticks to a Timed Loop on rising or falling edges of a digital signal. NI-DAQmx uses the **counter** you specify to detect the edges of the digital signal. Use this VI to base a timing source on an external digital signal or on an internal digital signal not included in the <u>Signal From Task</u> instance of the DAQmx Create Timing Source VI.



- **source** specifies the <u>terminal</u> to which you connect the digital signal you want to use for the timing source. A <u>DAQmx terminal</u> <u>constant</u> lists all terminals available on devices installed in the system. You also can specify a source terminal by wiring a string that contains a <u>terminal name</u>.
- **counter** specifies the name of the counter to use for the timing source. The <u>DAQmx physical channel constant</u> lists all physical channels, including counters, for devices installed in the system. You also can wire a string that contains a <u>name</u> of a counter to this input.
- **edge** specifies on which edges of the digital signal you want to send ticks to the Timed Loop.

Falling (10171)Send ticks on falling edges.Rising (10280)Send ticks on rising edges.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **timing source name** is the name to assign to the timing source this VI creates.
- edge count specifies the number of edges of the digital signal that must occur for the timing source to send a tick to the Timed Loop. For example, if you set this value to 3, the timing source sends a tick to the Timed Loop on every third edge of the digital signal.
- timing source out is the name of the timing source this VI creates. Wire this output to the **source** input of the Input Node of a Timed Loop. If you specified a name for the timing source in the **timing source name** input, this output contains the same name. If you did not specify a name for the timing source, NI-DAQmx creates a name and sets this output to that name.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the

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

Digital Change Detection

Creates a timing source that sends ticks to a Timed Loop on rising and/or falling edges of one or more digital lines. You can use this instance of this VI only with devices that support digital change detection. Use the <u>Digital</u> <u>Edge (Counter)</u> instance of the DAQmx Create Timing Source VI with other devices. That instance uses a counter and cannot detect edges on multiple lines.



- **rising edge physical channels** specifies the names of the digital lines or ports on which to detect rising edges. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in your system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input.
- **falling edge physical channels** specifies the names of the digital lines or ports on which to detect falling edges. The <u>DAQmx</u> <u>physical channel constant</u> lists all lines and ports for devices installed in your system. You also can wire a string that contains a <u>list or range</u> of digital lines or ports to this input.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE,

code is 0 or a warning code.

- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **timing source name** is the name to assign to the timing source this VI creates.
- timing source out is the name of the timing source this VI creates. Wire this output to the source input of the Input Node of a Timed Loop. If you specified a name for the timing source in the timing source name input, this output contains the same name. If you did not specify a name for the timing source, NI-DAQmx creates a name and sets this output to that name.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

Signal From Task

Creates a timing source that uses the **signal** you specify to determine when to send ticks to a Timed Loop. You must create a task that can generate the signal and wire that task to the **task/channels in** input of this VI. Do not use the <u>DAQmx Start Task</u> VI to start the task. The Timed Loop starts the task automatically.



- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- **signal** is the signal to use for the timing source.

Change Detection Event (12511)	Send a tick to the Timed Loop each time the Change Detection Event occurs.
Counter Output Event (12494)	Send a tick to the Timed Loop each time the Counter Output Event occurs.
Sample Clock (12487)	Send a tick to the Timed Loop on each active edge of the Sample Clock.
Sample Complete Event (12530)	Send a tick to the Timed Loop each time the Sample Complete Event occurs.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- timing source name is the name to assign to the timing source this VI creates.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **timing source out** is the name of the timing source this VI creates. Wire this output to the **source** input of the Input Node of a Timed Loop. If you specified a name for the timing source in the **timing source name** input, this output contains the same name. If you did not specify a name for the timing source, NI-DAQmx creates a name and sets this output to that name.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

DAQmx Timing Source Property Node

A Property Node with the <u>DAQmx Timing Source</u> class preselected. Right-click the Property Node and choose **Select Filter** from the shortcut menu to make the Property Node show only the properties supported by a particular device installed in the system or supported by all the devices installed in the system.

 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code**

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

- **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **property 1...n** is an example of a property you want to get (read) or set (write).

DAQmx Wait For Next Sample Clock

Waits until the next pulse of the <u>Sample Clock</u> occurs. If an extra Sample Clock pulse occurs between calls to this VI, the second call returns an error or warning and waits for the next Sample Clock pulse. Use the <u>Convert Late Errors to Warnings</u> DAQmx Real-Time property to specify whether this VI returns errors or warnings. If that property is TRUE, any warnings this VI returns do not include the **source** string.

Use this VI to ensure <u>I/O cycles</u> complete within Sample Clock periods. National Instruments recommends you use this VI for certain <u>applications</u> only.

Use the <u>DAQmx Real-Time</u> properties to configure error reporting and waiting options.



 \blacksquare Place on the block diagram. \blacksquare Find on the **Functions** palette.

- **task/channels in** is the name of the <u>task</u> or a list of <u>virtual</u> <u>channels</u> to which the operation applies. If you provide a list of virtual channels, NI-DAQmx creates a task automatically.
- timeout (sec)specifies the maximum amount of time in seconds to wait for the next Sample Clock pulse. If the time elapses, this VI returns an error. The default timeout is 10 seconds. If you set timeout to -1, this VI waits indefinitely.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. If an error occurs while this VI or function runs, the VI or function runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **task out** is a reference to the task after this VI or function runs. If you wired a channel or list of channels to **task/channels in**, NI-DAQmx creates this task automatically.
- **is late?** indicates if this VI detected an extra Sample Clock pulse after the specified number of warmup iterations execute. Use the <u>Number of Warmup Iterations</u> DAQmx Real Time property to specify the number of warmup iterations. This output is always FALSE until that number of loop iterations execute.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, error out describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

TIO Devices

You can set the following properties on TIO devices while the task is running. If a property is not listed here, you cannot set it while the task is running.

DAQmx Channel

<u>Counter Output:Pulse:Time:High Time</u> <u>Counter Output:Pulse:Time:Low</u> <u>Time</u>

Counter Output:Pulse:Frequency:Frequency

Counter Output:Pulse:Frequency:Duty Cycle

Counter Output:Pulse:Ticks:High Ticks

Counter Output:Pulse:Ticks:Low Ticks