



DAQ Assistant Help

January 2008, 370467L-01

This help file describes how to use the DAQ Assistant to interactively configure common measurement tasks, global virtual channels, or scales. In NI LabVIEW 7.x and later, NI LabWindowsTM/CVITM 7.x and later, or NI Measurement Studio 7.x and later, you also can use the DAQ Assistant to generate NI-DAQmx code from your tasks and global virtual channels. You can also use the DAQ Assistant with NI LabVIEW SignalExpress 2.x and later.

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

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

<u>Conventions</u>—formatting and typographical conventions used in this help file

Related Documentation

Glossary

Important Information

<u>Technical Support and Professional Services</u>

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

© Copyright 2003–2008 National Instruments Corporation. All rights reserved.

Related Documentation

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

- DAQ Getting Started Guide—This guide describes how to install the NI-DAQmx driver software, your data acquisition (DAQ) device, and how to confirm that your device is operating properly.
- LabVIEW Help—This help file contains information about LabVIEW palettes, menus, tools, VIs, and functions. This help file also includes step-by-step instructions for using LabVIEW features. In LabVIEW, select Help.
- LabWindows/CVI Help—The LabWindows/CVI Help includes the following sections:
 - Using LabWindows/CVI—Information about windows, menus, commands, dialog boxes, and options for customizing configuration defaults.
 - Library Reference—Reference information for all LabWindows/CVI library functions.
 - Programmer Reference—Information about developing programs in LabWindows/CVI.
 - Example Programs—Descriptions of each of the example programs included with LabWindows/CVI.
 - Tools Library—Descriptions of the additional instrument drivers included with LabWindows/CVI.

In LabWindows/CVI, select **Help»Contents**.

- Measurement & Automation Explorer Help for NI-DAQmx—This help file contains information on configuring and testing DAQ devices, SCXI devices, PXI systems, SCC devices, serial and parallel ports, and RTSI cables using Measurement & Automation Explorer (MAX) for NI-DAQmx, and information on special considerations for operating systems. In MAX, select Help»Help Topics»NI-DAQmx.
- NI Measurement Studio Help—This help file contains information about using Measurement Studio for Microsoft Visual C++, Visual

- Basic, and the .NET platform. In Visual Studio .NET, select **Help»Contents**.
- NI-DAQmx Help—This help file describes how to use NI-DAQmx to program your National Instruments device. NI-DAQmx is the software you use to communicate with and control your NI device. Select Start»All Programs»National Instruments»NI-DAQ.
- Taking an NI-DAQmx Measurement in LabVIEW—This tutorial teaches you how to take an NI-DAQmx measurement in LabVIEW 7.x or later using the DAQ Assistant. The DAQ Assistant is a graphical interface for configuring measurement tasks, global virtual channels, and scales. You also can use the DAQ Assistant to generate NI-DAQmx code from the task. Refer to Taking an NI-DAQmx Measurement in LabVIEW in the LabVIEW Help.

Using Help

Conventions

Navigating Help

Searching Help

Printing This Help File

Conventions

This help file uses the following conventions:

Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DIO<3..0>.

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence File»Page Setup»Options directs you to pull down the File menu, select the Page Setup item, and select Options from the last dialog box.

S

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

bold Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names, emphasis, or an introduction to a key concept.

dark red Text in this color denotes a caution.

<u>green</u> Underlined text in this color denotes a link to a help topic, help file, or Web address.

italic Italic text denotes variables, emphasis, cross references, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

Supply

monospace Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Navigating Help (Windows Only)

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

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

Searching Help (Windows Only)

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

Wildcards

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

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

Nested Expressions

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

Boolean Expressions

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

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

Search Options

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

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

Printing Help File Topics (Windows Only)

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

- 1. Right-click the book.
- 2. Select **Print** from the shortcut menu to display the **Print Topics** dialog box.
- 3. Select the **Print the selected heading and all subtopics** option.



Note Select **Print the selected topic** if you want to print the single topic you have selected in the **Contents** tab.

4. Click the **OK** button.

Printing PDF Documents

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

Getting Started

The DAQ Assistant helps you configure <u>tasks</u>, <u>virtual channels</u>, and <u>scales</u> using a graphical interface. You can launch the DAQ Assistant from your National Instruments application software such as LabVIEW, LabVIEW SignalExpress, LabWindows/CVI, or Measurement Studio. You also can launch the DAQ Assistant from MAX.

Using the DAQ Assistant, you can do the following things:

- Create and edit tasks and virtual channels
- Add virtual channels to tasks
- Create and edit scales
- Test your configuration
- Save your configuration
- Generate code in your NI application software for <u>use</u> in your application
- View connection diagrams for your sensors



Note You must use version 7.x or later of LabVIEW, LabWindows/CVI, or Measurement Studio or version 2.x or later of LabVIEW SignalExpress to use the DAQ Assistant.

Channels and Tasks

Physical Channels and Virtual Channels

A physical channel is a terminal or pin at which you can measure or generate an analog or digital signal. A virtual channel is a collection of settings such as a name, a physical channel, input terminal connections, the type of measurement or generation, and can include scaling information. In NI-DAQmx, virtual channels are integral to every measurement. In NI-DAQmx, use the DAQ Assistant, accessible from MAX or NI application software to configure channels and measurement tasks. Refer to the DAQ Assistant Help and Measurement & Automation Explorer Help for NI-DAQmx. You can also configure virtual channels with the NI-DAQmx API in your application program. In Traditional NI-DAQ (Legacy) and earlier versions, you can use MAX to configure virtual channels, an optional way to record which channels are being used for different measurements.

Tasks

A *task*, an important concept for NI-DAQmx, is a collection of one or more virtual channels with timing, triggering, and other properties. Conceptually, a task represents a measurement or generation you want to perform. You can set up and save all of the configuration information in a task and use the task in an application.

Global Virtual Channels and Local Virtual Channels

In NI-DAQmx, you can configure virtual channels as part of a task or separate from a task. Virtual channels created inside a task are *local virtual channels*. Virtual channels defined outside a task are *global virtual channels*. You can create global virtual channels in MAX or in your application software and save them in MAX. You can use global virtual channels in any application or add them to a number of different tasks. If you modify a global virtual channel, the change affects all tasks in which you reference that global virtual channel. In most cases, it is simpler to use local virtual channels.



Note If you are using LabVIEW 8.0 and later, you can also create tasks and channels local to a LabVIEW Project.

Using the DAQ Assistant in Your Application

Using the DAQ Assistant, you can <u>create</u>, <u>configure</u>, and <u>save virtual</u> <u>channels</u> or <u>tasks</u> for <u>use</u> in your applications. If you are using NI application software, you also can <u>generate code</u> based on a task or virtual channel.

Refer to LabVIEW Help, LabWindows/CVI Help, Measurement Studio Help, or NI-DAQmx Key Concepts for more information about programming with your task or global channel.

Opening the DAQ Assistant

You can open the DAQ Assistant from your NI application software or from MAX. When you open the DAQ Assistant, a wizard prompts you for the information needed to create your new <u>task</u>. When the wizard is done, the DAQ Assistant opens, and you can configure timing, triggering, measurement-specific settings, and scaling.

You can also use the DAQ Assistant to create a global virtual channel.

You can open the DAQ Assistant to create a task in:

- LabVIEW
- LabWindows/CVI
- Measurement Studio
- MAX

You can also use the DAQ Assistant in <u>LabVIEW SignalExpress</u> to create DAQ steps or to use global virtual channels in your project.

Opening the DAQ Assistant in LabVIEW

You can open the DAQ Assistant to create a task in one of the following ways:

- Place the DAQmx Task Name control on the front panel. Rightclick the control, and select New DAQmx Task.
- If you are using LabVIEW 8.0 and later, you can create a task local to the Project, or you can create a task in MAX. To create a task in a Project, right-click **My Computer** in the Project Explorer window. Then, select **New»Task**.
- In LabVIEW 7.x, click the Configure pull-down menu on the main LabVIEW launch screen and select Create New Task (DAQ Assistant).
- Place the DAQ Assistant Express VI on the block diagram. This VI creates tasks that are fully contained in a single VI.



Note If you configure a task using the DAQ Assistant Express VI, the task is local to the application and cannot be saved to MAX for use in other applications. You must convert the DAQ Assistant Express VI to an NI-DAQmx Task to use the task in other applications. You can also generate code from the DAQ Assistant Express VI. Right-click the DAQ Assistant Express VI on the front panel in LabVIEW and select **Generate NI-DAQmx Code**.

• In LabVIEW 7.x or later, select **File»New** and browse the From Templates folder to **Data Acquisition with NI-DAQmx**.

Refer to *Taking an NI-DAQmx Measurement in LabVIEW* in the LabVIEW Help for step-by-step instructions for creating a task in LabVIEW and generating code based on the task.

For more information about creating tasks and channels with Project in LabVIEW, refer to the LabVIEW Help.

After opening the DAQ Assistant, <u>create</u> the new task.

Opening the DAQ Assistant in LabWindows/CVI

To open the DAQ Assistant, select Tools»Create/Edit DAQmx Tasks.

In the Create/Edit DAQmx Tasks dialog box, you can choose to create a task in MAX or in the LabWindows/CVI project. You must use MAX to <u>create global virtual channels</u> for use in LabWindows/CVI. For information about the differences between creating a task in the project or in MAX, refer to the *LabWindows/CVI Help*.

You also can create a MAX-based task in LabWindows/CVI when you select **New Task** from the **Task Name** control in the DAQmxLoadTask function panel.

After opening the DAQ Assistant, <u>create</u> the new task.

Opening the DAQ Assistant in MAX

To open the DAQ Assistant, complete the following steps:

 Right-click **Data Neighborhood**, and select **Create New** from the shortcut menu.

If you are using a remote system running LabVIEW Real-Time, expand **Remote Systems**, find and expand your target, then right-click **Data Neighborhood** and select **Create New**.

- 2. Select **NI-DAQmx Task** in the Create New window.
- 3. Click **Next**. The DAQ Assistant opens.
- 4. <u>Create</u> your task.
- 5. Click **Save Task** to save the task to MAX.



Tip You can also right-click your device under Devices and Interfaces and select **Create New** to launch the DAQ Assistant from MAX.

Opening the DAQ Assistant in Measurement Studio

To use a DAQmx task class in a Measurement Studio application, you must first create the DAQmx task class within the application. You must use MAX to <u>create global virtual channels</u> for use in Measurement Studio. Complete the following steps to create a DAQmx task class:

- 1. Open the project in which you want to create a DAQmx task class.
- Select Project»Add New Item to open the Add New Item dialog box.
- 3. In the Categories pane, select **Measurement Studio»Assistants**.
- 4. In the Templates pane, select **DAQmx Task**.
- 5. Specify a name for the DAQmx task file and click **Open**. The name you choose for the file becomes the class name.
- 6. In the New DAQmx Task dialog box, specify whether to create a new task or to make a project copy of a global task that you create in MAX.



Note If you choose a global task and make changes to the task, those changes are not saved to the global task stored in MAX.

7. Click **Finish**. The DAQ Assistant opens for you to <u>create</u> the new task.

After you complete these steps, the DAQ Assistant adds a .mxb file to the project. The .mxb file stores the DAQmx task class configuration information. The DAQ Assistant adds references to appropriate NI class libraries to the project.

Global Virtual Channels

To open the DAQ Assistant to create global virtual channels, complete the following steps:



Note If you are using LabWindows/CVI or Measurement Studio, you must create your global virtual channels in MAX.

- 1. In MAX, right-click **Data Neighborhood** and select **Create New**. If you are using a remote system running LabVIEW Real-Time, expand **Remote Systems**, find and expand your target, then right-click **Data Neighborhood** and select **Create New**.
- 2. In the Create New window, select **NI-DAQmx Global Virtual Channel** and click **Next**. The DAQ Assistant opens.

You also can open the DAQ Assistant directly within LabVIEW, LabWindows/CVI, or Measurement Studio version 7.x or later:

- In LabVIEW and LabVIEW Real-Time, use the DAQmx Global Channel control to open the DAQ Assistant. Rightclick the control and select New Channel (DAQ Assistant).
- If you are using LabVIEW 8.0 or later, you can create channels local to a LabVIEW Project, or you can create channels in MAX. To create channels in a Project, right-click My Computer in the Project Explorer window. Then, select New»Channel.
- 3. In LabVIEW 7.x, click the **Configure** pull-down menu and select **Create DAQmx Channel (DAQ Assistant)**.
- 4. In LabWindows/CVI or Measurement Studio, first you must create a task as previously described. In the DAQ Assistant, click **Add Existing DAQmx Global Channels** to add global virtual channels to your task. You must use MAX to create global virtual channels.
- 5. Configure the global virtual channel.
- 6. <u>Save</u> your configuration.
 - If you opened the DAQ Assistant from LabVIEW or LabWindows/CVI, click OK.
 - If you opened the DAQ Assistant from Measurement Studio, select **File**»**Save**.

• If you opened the DAQ Assistant from MAX, click **Save Channel**.

You now can use the global virtual channel in an application or add the global virtual channel to a task. Refer to the application software documentation for more information about <u>using a global virtual channel</u> or <u>generating code</u>.

Using the DAQ Assistant in LabVIEW SignalExpress

You can use the DAQ Assistant with LabVIEW SignalExpress to create DAQ steps. To open the DAQ Assistant in LabVIEW SignalExpress, complete the following steps:

- 1. Click Add Step.
- 2. Select either Acquire Signals»DAQmx Acquire or Generate Signals»DAQmx Generate.
- 3. Select the I/O type, such as analog input, for your measurement.
- 4. Select the measurement type, such as voltage.
- 5. The DAQ Assistant opens, beginning with the **Configuration** tab. You can configure settings and <u>test</u> your step.

Creating a Task or Global Virtual Channel

After the DAQ Assistant <u>opens</u>, complete the following steps to create a new task or global virtual channel:

- 1. Select an I/O type, such as analog input.
- Select the measurement or generation to perform. You can later add <u>multiple measurement types</u> to a task, but you must first create the task with one measurement type.
- 3. Select the sensor to use, if applicable for that measurement.
- 4. A dialog box for the task you specified prompts you to select the physical channels from which to create local virtual channels in the task. Alternatively, you can add global virtual channels to the task, or copy information from an existing global virtual channel to local virtual channels.

If your system is configured with a TEDS sensor, click the **TEDS** tab. Select the physical channel(s). The DAQ Assistant imports the setting from the sensor, and uses the information to create the task. Click **Next**.



Note For hardware that supports multiple channels in a task, you can select <u>multiple channels</u> to add to your task at the same time.

- 5. Type the new task or global virtual channel <u>name</u>. Click **Finish**.
- 6. Configure measurement-specific settings, timing, triggering, and scaling, if necessary. If one or more channels are configured for TEDS sensors, some settings loaded from TEDS are not editable.
- 7. Save your configuration.



Note In a task, you can <u>add channels</u> of the same or different measurement types.

Adding Multiple Channels to Tasks

For hardware that supports multiple physical channels in a task, you can select multiple channels to add to the task at the same time. To select particular physical channels, hold the <Ctrl> key while selecting the channels. To select a range of physical channels, click the first physical channel, then press and hold the <Shift> key, and select the last physical channel in the range. All physical channels between the two selections are highlighted.

You can also select multiple global virtual channels and any TEDS channels you have configured.

When you click the Add Channels button, a list of all channels of the same I/O type as your task appears. This list includes channels from all devices in your system. Unless your hardware supports <u>multidevice</u> <u>tasks</u>, you can add only channels from the same device to the task.

You can also create a task with multiple channels of different measurement types.

Duplicating NI-DAQmx Global Virtual Channels

To copy a global virtual channel, complete the following steps:

- 1. Expand **Data Neighborhood**.
- 2. Expand NI-DAQmx Global Virtual Channels.
- 3. Right-click the global virtual channel you want to copy.
- 4. Select **Duplicate**.
- 5. A dialog box prompts you to select physical channels from a list of valid channels for the measurement type. Select the physical channels to use in the new global virtual channels. The DAQ Assistant generates an ID for each new global channel, consisting of the original ID with an incremented number appended to the end.
- 6. Click OK.

Creating a Task with Multiple Measurement Types

To create a task with multiple measurement types, first create a task with an initial measurement type. Then, click the **Add Channels** button to add the next channel(s).



Note You can have multiple measurement types, such as voltage, temperature, and pressure, in the same task. However, your task must contain only one measurement class, such as analog input or analog output.

When you click the Add Channels button, a list of all channels of the same I/O type as your task appears. This list includes channels from all devices in your system. Unless your hardware supports <u>multidevice</u> tasks, you can add only channels from the same device to the task.

Naming Channels, Tasks, and Scales

Use the following guidelines to name your channels, tasks, and scales:

- Use any alphanumeric characters.
- Do not use non-alphanumeric characters with the following exceptions:
 - In NI-DAQmx 7.4 or later, dashes are allowed in channel, task, and scale names.
 - Spaces are allowed.
 - You can use underscores within the channel, task, or scale name, but you cannot use leading underscores, such as Dev1.
- You must use no more than 256 characters.

Special Considerations with LabVIEW Project

In the DAQ Assistant, you receive a warning dialog if you attempt to name a LabVIEW Project channel the same name as an existing MAX global virtual channel. The DAQ Assistant forces you to choose a unique name. However, in LabVIEW Project, you can right-click a channel and rename it to the same name as a MAX global virtual channel. If you do rename the channel to the same name as a MAX channel, the next time you use the DAQ Assistant, or an I/O Control, the MAX channel will be grayed out and you cannot select it.

Editing a Task or Global Virtual Channel

You can use tasks and global virtual channels in your NI application software or in MAX.

LabVIEW

Complete the following steps to edit a task or global virtual channel saved in MAX:

- 1. Place either a DAQmx Task Name control or a DAQmx Global Channel control on the front panel.
- 2. Select the task or global virtual channel from the pull-down list.
- 3. Right-click the control and select **Edit DAQmx Task** or **Edit DAQmx Channel**. Then, select the destination.
- 4. Modify the task or global virtual channel when the DAQ Assistant launches.
- 5. Click **OK** to save the changes.

LabVIEW SignalExpress

With LabVIEW SignalExpress 2.5 and later, you can import tasks from MAX and copy global virtual channels from MAX into DAQ steps. If you import a task from MAX, the task becomes local to that LabVIEW SignalExpress project, and you cannot save it back to MAX. Also, any global virtual channels contained in the task become local virtual channels. Then, you can <u>use the channels</u> in your LabVIEW SignalExpress application.

LabWindows/CVI

You must use \underline{MAX} to edit global virtual channels.

To edit a task, select **Tools»Create/Edit DAQmx Tasks** and choose the task you want to edit. You also can edit MAX-based tasks within MAX by selecting **Data Neighborhood»NI-DAQmx Tasks** and choosing the task you want to edit.

Measurement Studio

You must use $\underline{\mathsf{MAX}}$ to edit global virtual channels or MAX-based tasks.

To edit a project-based task, go to the project and double-click the .mxb file to launch the DAQ Assistant.

MAX

To edit a task or global virtual channel using the DAQ Assistant from within MAX, complete the following steps:

- 1. Expand Data Neighborhood.
- 2. Expand NI-DAQmx Tasks or NI-DAQmx Global Virtual Channels.
- 3. Click the task or global virtual channel you want to edit.
- 4. Modify the task or global virtual channel in the DAQ Assistant.
- 5. Click **Save** when you are finished.



Note If an asterisk appears next to a global virtual channel, the channel has unsaved changes.

Adding Virtual Channels to Tasks

You can add virtual channels to your task. Additional virtual channels must be of the same class, such as analog input or analog output, as the task. To add virtual channels to your task, complete the following steps:

- 1. Click the Add button.
- 2. From the drop-down list, select the measurement type to add.
- 3. A dialog box prompts you to select a physical channel from a list of valid physical channel(s) for that measurement type. Select the physical channels to add to your task. Alternatively, you can add existing global virtual channels or copy information from an existing global virtual channel to a local virtual channel.
- 4. Click OK.
- 5. Repeat these steps to add more virtual channels.

Creating and Editing Scales

You can use the DAQ Assistant to <u>create</u> and <u>edit</u> scales for your <u>tasks</u> and <u>global virtual channels</u>. You can use an NI-DAQmx Scale to specify a conversion from the prescaled units measured by a physical channel to the scaled units associated with your transducer or actuator. You can then associate this scale with one or more virtual channels.

Often, you do not need to create an NI-DAQmx Scale, because NI-DAQmx has explicit support for the most common transducers and actuators. For example, when creating an analog input temperature virtual channel, you can specify which type of transducer (such as a thermistor, RTD, or thermocouple) is used to make the measurement. However, if NI-DAQmx does not explicitly support your transducer or actuator, you can create an NI-DAQmx Scale that specifies how to convert from the prescaled units to the scaled units. Notice that the same NI-DAQmx Scale can be associated with multiple virtual channels. You do not need to create an NI-DAQmx Scale for each virtual channel if the scale is the same.

Creating Scales

Follow these steps to create new scales:

- 1. You can create a new scale in the following ways:
 - DAQ Assistant—Click the Scale icon and select Create New.
 - MAX—Right-click Scales in the configuration tree, and select New Scale. In the Create New window, select NI-DAQmx Scale.
 - LabVIEW—Place the DAQmx Scale Name control on the front panel. Right-click the control and select New Scale (DAQ Assistant).
- 2. When the Create Scale wizard launches, select the scale type and click **Next**.
- 3. Enter a name for your scale.
- 4. Click **Finish** to continue configuring your scale in the DAQ Assistant.
- 5. Click **OK** or **Save Scale** to save your scale to MAX.

Editing Scales

LabVIEW—Complete the following steps to edit a previously saved custom scale:

- 1. Place a DAQmx Scale Name control on the block diagram.
- 2. Select a custom scale you previously created and saved.
- 3. Right-click the control and choose Edit Scale (DAQ Assistant).
- 4. Modify the scale in the DAQ Assistant.
- 5. Click **OK** to save the changes.

MAX—Complete the following steps to edit a previously saved custom scale:

- 1. Expand Scales.
- 2. Click the scale you want to edit.
- 3. Modify the scale in the DAQ Assistant.
- 4. Click **Save Scale** when you are finished.

Testing Your Configuration

Click **Run** to test your configuration and make sure your sensors and signals are connected properly. Data acquired appears in the preview view. Verify that you are acquiring the expected data and that you connected the signal properly. If necessary, modify any settings and click **Run** to test your configuration again.

Saving Your Configuration

You can save your task or global virtual channel configuration to MAX in several ways, depending on how you launched the DAQ Assistant. To save your configuration, do the following:

- MAX—Click Save on the toolbar. You also can check for errors in your task by clicking Save. If your task has an error, a dialog box appears. Click No in the dialog box to cancel the save and show the error(s).
- LabVIEW—Click OK.



Note If you configure a task using the DAQ Assistant Express VI, the task is local to the application and cannot be saved to MAX for use in other applications. You must convert the DAQ Assistant Express VI to an NI-DAQmx Task to save the task to MAX for use in other applications or to generate code.

- LabWindows/CVI—Click OK.
- Measurement Studio—Select File»Save.

You can access saved global virtual channels and tasks in Data Neighborhood in MAX or <u>in your NI application software</u>.



Note You cannot save a global virtual channel or a task to MAX with LabVIEW SignalExpress.

Using Your Task or Global Virtual Channel in Your Application Software

You can use <u>tasks or global virtual channels</u> saved in MAX when programming your applications.

If you are using LabVIEW 7.x or later, LabWindows/CVI 7.x or later, or Measurement Studio 7.x or later, you can generate code from your task or global virtual channel for use in your application.

Refer to LabVIEW Help, LabWindows/CVI Help, Measurement Studio Help, or NI-DAQmx Help for more information about programming with your task or global virtual channel. Refer to LabVIEW SignalExpress Help for information about using the DAQ Assistant in LabVIEW SignalExpress.



Note You can access global virtual channels in LabWindows/CVI and Measurement Studio only if you are adding them to a task.

LabVIEW

Complete the following steps to access your task or global virtual channel:

- 1. Place either the DAQmx Task Name control or the DAQmx Global Channel control on the front panel.
- 2. Click the pull-down menu and select the task or global virtual channel(s) you want to use in your application. You can select multiple global virtual channels, but not multiple tasks. You also can select **Browse** to find tasks and global virtual channels.

LabVIEW SignalExpress

To access a task you created in MAX, select **Tools»Import Task from MAX** in LabVIEW SignalExpress 2.5 and later.



Note Any global virtual channels in the task you import become local virtual channels in LabVIEW SignalExpress.

You also can copy global virtual channels into your DAQ steps in LabVIEW SignalExpress when you are adding channels to the step.

- 1. Select the **Virtual** tab. The Virtual tab appears only if you have already created a global virtual channel of the same measurement type as your step.
- 2. Select the global virtual channel you want to copy to the step. When you copy the global virtual channel to your step, it becomes a local virtual channel.

LabWindows/CVI

Complete the following steps to access your task:

- 1. Open the DAQmxLoadTask function panel.
- 2. Press <Enter> in the **Task Name** control. A list of task names you can use appears in the context menu.
- 3. Select the task you want to use from the context menu.

Measurement Studio

Launch one of the following wizards from the **Project»Add New Item** menu:

- To use a task stored in MAX to create a new project-based task, use the DAQmx Task Add New Item wizard.
- To reference a task stored in MAX when creating a Visual C++ DAQmxUserCode, use the DAQmxUserCode Add New Item wizard.
- To reference a task stored in MAX when creating a .NET DAQmxUserControl, use the DAQmxUserControl Add New Item wizard.

Each of these wizards opens a dialog box you can use to select a task saved in MAX.

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

When you configure a task using the DAQ Assistant Express VI, 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 if you want to save the task to MAX for use in other applications or to generate code.



Note If you are working within a project in LabVIEW, 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.

- 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. An NI-DAQmx Task replaces the DAQ Assistant Express VI.

Calibrating Channels

Use the Channel Calibration Wizard to add software channel calibration to a virtual channel. Channel calibration takes sensors, hardware, and measurement software into account and applies compensation to the measurement in addition to all scaling and hardware calibration, thus improving measurement accuracy.

Complete the following steps to launch the Channel Calibration Wizard from the DAQ Assistant and calibrate analog input virtual channels.

- Select an existing task or global channel or create a new task or global channel.
- 2. Click the **Calibration** tab in the DAQ Assistant.
- 3. Select one or more virtual channels from the Channel List. You must select only one type of virtual channel. If you select virtual channels of different measurement types, the Calibrate button is disabled.

4. Click Calibrate.

The Channel Calibration Wizard launches. Complete the screens to calibrate the selected virtual channel. You can calibrate as many data points as needed to produce a calibration table.

The Channel Calibration Wizard stores a set of calibration properties for each virtual channel you calibrate. Select **Properties** in the Calibration tab of the DAQ Assistant to view calibration history, verify the calibration, recalibrate the selected virtual channel(s), and generate a calibration report. If you select virtual channels of different measurement types, the Properties button is disabled.

After calibrating the virtual channel, click **Enable Calibration** to enable or disable the calibration. If Enable Calibration is disabled, the calibration is saved but not applied to the selected virtual channels. To delete a calibration, select one or more virtual channels in the Channel List and click **Delete Calibration**.

In the Calibration tab, the Calibration Date and Expiration Date show when the last calibration occurred and when the calibration expires. If these indicators are disabled, you have selected virtual channels that have different calibration and/or expiration dates or are not calibrated.

Generating Code

If you are using NI application software version 7.x or later, you can generate code for use in your applications. You can generate code in LabVIEW 7.x or later, LabVIEW SignalExpress 2.x or later, LabWindows/CVI 7.x or later, and Measurement Studio 7.x or later from any task saved in MAX. You also can generate code from a global virtual channel in LabVIEW 7.x or later.

Any changes you make to the generated code apply only to the task in the application and are not saved back into the task or global virtual channel in the DAQ Assistant.

Types of Code Generation

There are three options for generating code in your NI application software from a task or global virtual channel:

- Example—Generates all the code necessary to run the task or global virtual channel, such as VIs needed to read or write samples, VIs to start and stop the task, loops, and graphs. Choose this option if you want to run the task or global virtual channel you created to verify that it works or to use your configuration in a simple application. The generated code is a simple NI-DAQmx example that you can then modify for your application.
- Configuration—Generates the code that replicates the configuration of the tasks and global virtual channels. Choose this option if you want to deploy your application to another system.
 Refer to Deployment in the NI-DAQmx Help located at Start»All Programs»National Instruments for more information.

When you generate **Configuration** code, the link between the application and the DAQ Assistant is lost. Any changes you make to the configuration code are not reflected in the DAQ Assistant. You can regenerate **Configuration** code from the DAQ Assistant, but the regenerated code does not maintain previous changes that you made to the code.



Caution Although generated configuration code includes code to create global virtual channels and tasks, it does not include code to create scales. If you generate configuration code for deployment of applications that include scales, you must ensure that the scales are configured in the target computer. Refer to *Deployment* in the *NI-DAQmx Help* for deployment options.

• **Configuration and Example**—You must use LabVIEW to generate this type of code. Choose this option if you want to create a portable, complete application that can run on other systems.



Note If you right-click the DAQ Assistant Express VI and select **Generate NI-DAQmx Code**, you generate both configuration and example code.

Generating Code in Your Application Software

Select your NI application software for more information on the types of code you can generate and how to use the DAQ Assistant to generate code.

- <u>LabVIEW</u>
- <u>LabVIEW SignalExpress</u>
- LabWindows/CVI
- Measurement Studio

Generating Code in LabVIEW

This section describes what happens during code generation in LabVIEW and how to generate code using the DAQ Assistant.

Types of Code Generation

You can generate all three types of code in LabVIEW. For general information about the types of code and special considerations, refer to <u>Generating Code</u>.

- **Example**—This option adds to the VI you are working in.
- Configuration—The I/O constant/control is replaced with a subVI that contains VIs and property nodes used for physical channel creation and configuration, timing configuration, and triggering configuration used in the task or global virtual channel.



Caution Although generated configuration code includes code to create global virtual channels and tasks, it does not include code to create scales. If you generate configuration code for deployment of applications that include scales, you must ensure that the scales are configured in the target computer. Refer to *Deployment* in the *NI-DAQmx Help* for deployment options.

• Configuration and Example—This option generates both Configuration code and Example code for the task or global virtual channel in one step.

Generating Code in LabVIEW

Complete the following steps to generate code:

- 1. Place either the DAQmx Task Name control or DAQmx Global Channel control on the front panel.
- 2. Click the pull-down menu and select the task or global virtual channel from the list.
- 3. Right-click the control and select **Generate Code»Example**, **Configuration**, or **Configuration and Example**.

Generating Code in LabVIEW Using the DAQ Assistant Express VI

If you configure a task using the DAQ Assistant Express VI, you can right-click the VI and select **Generate NI-DAQmx Code** to generate configuration and example code. The code replaces the DAQ Assistant Express VI.



Note When you configure a task using the DAQ Assistant Express VI, the task is local to the application and cannot be saved to MAX for use in other applications. You can <u>convert the DAQ Assistant Express VI to an NI-DAQmx Task</u> to save the task to MAX for use in other applications.

Generating Code in LabVIEW SignalExpress

After you create and save a project in LabVIEW SignalExpress, you can generate LabVIEW code from LabVIEW SignalExpress. Refer to the *LabVIEW SignalExpress Help* for instructions on how to generate VIs from a LabVIEW SignalExpress project.

If you have DAQ steps in your LabVIEW SignalExpress project, those steps are converted to DAQ Assistant Express VIs in LabVIEW.

However, in the following situations, LabVIEW SignalExpress creates a top-level Express VI for your entire project:

- You have synchronized DAQ steps by sharing triggering or timing signals between them in one project.
- Any DAQ step contains more than one measurement type, such as voltage and temperature.
- DAQ steps are analog output or digital output.
- You have configured execution delays between your DAQ steps.

In these cases, you must double-click the top-level express VI to open LabVIEW SignalExpress to change the project settings.

Generating Code in LabWindows/CVI

This section describes what happens during code generation in LabWindows/CVI and how to generate code using the DAQ Assistant.

Types of Code Generation

You can generate two types of code in LabWindows/CVI. For general information about the types of code and special considerations, refer to Generating Code.

- Example—This option creates a function that has the task and adds the source files to the project.
- Configuration—The DAQmxLoadTask function is replaced with a call to a DAQ Assistant-generated function that builds the task programmatically.



Caution Although generated configuration code includes code to create global virtual channels and tasks, it does not include code to create scales. If you generate configuration code for deployment of applications that include scales, you must ensure that the scales are configured in the target computer. Refer to *Deployment* in the *NI-DAQmx Help* for deployment options.

Generating Code in LabWindows/CVI

To generate example code in LabWindows/CVI, complete the following steps:

- 1. Right-click a call to DAQmxLoadTask in a source file.
- 2. Select **Generate DAQ Example Code**.
- 3. Specify the name of the function that runs your task and the file to contain the generated example code.
- 4. Click OK.

The generated code includes source and header files that define the run task function. LabWindows/CVI automatically adds these files to the project and modifies the current source file to call the run task function. When called, the run task function programmatically creates a user interface panel that allows you to start the task and display the data.

To generate configuration code, you can store a task in your project, which causes the DAQ Assistant to generate source code to create the specified task programmatically. The generated source and header file define an entry point function that configures the task and returns a task handle. A generated binary file (.mxb) contains a binary description of the task that is used when editing the task in the DAQ Assistant. The DAQ Assistant adds all three files to the project when you create a project-based task. You can convert a MAX-based task to a project-based task by right-clicking DAQmxLoadTask in a source file and selecting **Copy DAQ Task To Project** from the context menu. This option performs the following actions:

- Generates and adds files to the project
- Replaces DAQmxLoadTask with a call to the generated entry point function
- Adds an include statement for the generated header file

Generating Code in Measurement Studio

This section describes what happens during code generation in Measurement Studio and how to generate code using the DAQ Assistant.

Types of Code Generation

You can generate two types of code in Measurement Studio. For general information about the types of code and special considerations, refer to Generating Code.

- Example—This option creates a function that has the task and adds the source files to the project.
- Configuration—The DAQmxLoadTask function is replaced with a call to a DAQ Assistant-generated function that builds the task programmatically.



Caution Although generated configuration code includes code to create global virtual channels and tasks, it does not include code to create scales. If you generate configuration code for deployment of applications that include scales, you must ensure that the scales are configured in the target computer. Refer to *Deployment* in the *NI-DAQmx Help* for deployment options.

Generating Code in Measurement Studio

After you configure and save a task, a class is generated and added to the project. The class contains the source code that corresponds to the configuration information that is stored in the .mxb file.



Note When you use the DAQ Assistant configuration editor to make changes to the DAQmx task class, the .mxb file is saved and the task code is regenerated. If you make changes directly to the generated source code file, those changes are lost when the source code for the .mxb file is regenerated. To enhance or extend the code that the DAQ Assistant generates, you can copy the source code file, change the name of the DAQmx task class, and add the copied file to the project. The copied file is not associated with a .mxb file. As a result, the DAQ Assistant does not overwrite the source code file.

Generating Code for a .NET DAQmx Task Class

The DAQmx task class code is contained in a hidden .cs or .vb source code file in the project. This source code file has the same base name as the .mxb file, but has an extension that corresponds to the language of the source code. Complete the following steps to view the hidden .cs or .vb file:

- 1. Select **Show All Files** in the Solution Explorer toolbar.
- 2. In the Solution Explorer, expand the .mxb file node.
- 3. Right-click the .cs or .vb file and select **View Code** to view the code.

To generate the source code, the .mxb file has a **CustomTool** property that associates the DAQ Assistant with the .mxb file.



Tip You can view the property settings for the .mxb file. Select the file in the Solution Explorer and view its properties in the Properties Window (**View»Properties Window**).

Generating Code for a Visual C++ DAQmx Task Class

The DAQmx task class code is contained in a .h and .cpp file in the project. These files have the same base name as the .mxb file, but have different extensions for the header and source files.

The Measurement Studio Add-In monitors all changes to the .mxb file. When an .mxb file that is contained in a Visual C++ project is saved, the add-in causes the DAQ Assistant to generate the source code for the configuration that is stored in the .mxb file.

Refer to the *NI Measurement Studio Help* for more information about generating code and using a task in Measurement Studio.

Generating Diagrams and Reports

In the DAQ Assistant, you can use the <u>Connection Diagram</u> tool to display either a text or an HTML file of the connections from each sensor to the connector blocks.

You can also use the <u>MAX Report Wizard</u> to generate a report of the NI-DAQmx hardware and software configuration of the system.

Connection Diagrams

The Connection Diagram tool in the DAQ Assistant can create either text or HTML reports of the connections from each sensor to the connector block.

Viewing Connection Diagrams

To view a connection diagram from within the DAQ Assistant, click **Connection Diagram**. Select each virtual channel in the task to view connections from sensor to connector block. The connections list displays each of the connections in text format, specifying terminal names and numbers.

Use the Save as HTML option in the Connection Diagram tool to generate an HTML report of your task. The report contains the following items:

- List of virtual channels in the task
- Physical channels
- Device types
- Measurement types
- Connection diagrams

To save a connection diagram, click **Save as HTML** and enter a name for the HTML file. An images subdirectory is created that contains all of the connection diagram images. Use a web browser such as Microsoft Internet Explorer to display and print the HTML report.

Connection Diagram Support

The tool currently supports analog input measurement types and the following MIO and SCXI screw terminal blocks:

- CB-11
- CB-27
- CB-37FH
- CB-37FV
- CB-50
- CB-68LP
- CB-68LPR
- CB-C68
- DAQPad-6015/6016
- PXI-4204
- PXI-4220
- SCB-100
- SCB-68
- SCXI-1300, 1303, 1304, 1308, 1313, 1314, 1314T, 1315, 1317, 1320, 1321, 1322, 1327, 1328, 1338
- TB-2705
- TB-2706
- TB-2725
- TBX-68

MAX Report Generation

In addition to the <u>Connection Diagram</u> tool in the DAQ Assistant, you can use the <u>MAX Report Wizard</u> to generate a report of the NI-DAQmx hardware and software configuration of the system. The report includes the following:

- Real and simulated device information, including device name, device type, serial number, connected accessory, and other settings
- Task information, including channels, timing, triggering, and other property settings
- Global channel property settings
- Custom scale property settings

The MAX Report Wizard supports both local and remote systems.

Generating Current or Voltage

You can generate two main kinds of signals for channels:

- Single samples, including DC signals—When generating single samples, you can use software or hardware timing (if available) to control when your device generates a signal.
- **Multiple samples**—Use multiple samples to generate an AC signal, such as a sine wave. Also called buffered analog output, generating multiple samples involves the following steps:
 - 1. Your application writes multiple samples into a buffer.
 - 2. All the samples in the buffer are then sent to your device according to the timing you specify. You can use software or hardware timing (if your device supports hardware timing) to control when your device generates a signal.



Note Some current generation devices, such as the SCXI-1124, allow you to use either an internal or external current supply. The choice of supply is determined by the terminal connections used to connect the load to the device. If you plan to use an internal supply, connect the load to the terminals SUPPLY and ISINK. If you plan to use an external supply, join the supply and load to ISINK and GND. It is important to verify that the terminal connections are correct. In addition to proper connections, verify that the attached load does not exceed the capabilities of the device. Refer to your device documentation for more information.

Click here for more information about generating voltage.

Click here for more information about generating current.

Measuring Acceleration with an Accelerometer

Acceleration is a change in velocity with respect to time. Accelerometers are transducers that represent acceleration as a voltage. They are specified by the range, frequency response, and the sensitivity of the device.

Accelerometers come in two axial types. The most common accelerometer <u>measures acceleration</u> along only a single axis. The second type is a tri-axial accelerometer, which can create a 3D vector of acceleration.

Both types of accelerometer can be either passive or active sensors. Both types also come with either both leads insulated from the case (isolated) or with one lead grounded to the case. To measure acceleration with a passive sensor, the sensor must be connected to a charge-sensitive amplifier.

The active accelerometer is nearly identical to the passive accelerometer, with the addition of a charge-sensitive amplifier built inside the accelerometer. The current source provides the excitation for the built-in amplifier of the sensor, while the instrumentation amplifier measures the voltage potential across the leads of the sensor.

Some analog input devices support programmable AC or DC coupling. If your device supports this feature, your acceleration measurement task is created using AC coupling. If you desire a different coupling behavior, you can change this property programmatically.

Measuring Analog Frequency

To measure the frequency of an AC signal, you need circuitry that produces triggers of the same frequency as the measured signal. The parameters that determine when to trigger are **Threshold Level** and **Hysteresis**. Every time the signal passes from (**Threshold Level** – **Hysteresis**) to **Threshold Level**, a trigger occurs.



Note (Threshold Level – Hysteresis) must be between –0.5 and 4.48 for the SCXI-1126 module.

A pulse generator uses these triggers and produces a pulse once every frequency cycle. The width of this pulse is set by the input frequency range. As the input frequency range increases, the pulse width grows smaller.

This pulse train is then converted to a DC signal that has a level proportional to the duty cycle of the pulse train. The duty cycle is the fraction of a period of the pulse train when the pulse is occurring. The DC signal has a voltage that is proportional to the input frequency and can therefore be scaled to that frequency value.

Measuring Angular Displacement with an RVDT

Angular displacement is movement about an axis, such as the angular motion of the shaft of a motor. An angular displacement sensor is a device whose output signal represents the rotation of the shaft; it cannot measure the physical displacement of the whole shaft. One type of sensor that measures angular displacement is a rotary variable differential transformer (RVDT).

RVDTs are the rotational version of LVDTs and generally operate over a range of ±30°–70°. They are available in servo-mount and can rotate through 360° without encountering a stop.

RVDTs require special electronics designed for the sensor. RVDTs typically have a delay of 10 ms caused by required filtering in the signal conditioner. They are extremely rugged and operate over wide temperature ranges. In environments characterized by extremes in temperature and shock, an RVDT is the clear choice for rotational applications when you need more than 70° of measurement range.

Measuring Current

Most measurement devices can measure voltages within a certain range. With a resistor, you also can measure the current through an analog input connector. To do so, you must place a known resistance in parallel with the input connector and current source.

You can measure voltage dropped across the resistor and convert it to current using Ohm's Law:

$$I(A) = V(V) / R(\Omega)$$

where I is the current, V is the voltage, and R is the resistance.

Measuring current is common because many devices generate a 4–20 mA current to represent a physical quantity. For instance, you can use current to measure the level in a tank. Using a level sensor, 4 mA could represent an empty tank, and 20 mA could represent a full tank.

Measuring Custom Voltage with Excitation

This measurement type allows you to use sensors that are not explicitly supported by NI-DAQmx and require voltage excitation, such as bridge sensors, load cells, and pressure transducers. Typically, you use this measurement type with SCXI modules that include an excitation source and configure a custom scale to convert the voltage the sensor returns to the units you want to measure.

For sensors that are not explicitly supported by NI-DAQmx and do not require any excitation, use the voltage measurement type with a custom scale. For sensors that are not supported by NI-DAQmx and require current excitation, use the voltage measurement type with a custom scale, and use the AI excitation DAQmx Channel properties to configure the excitation.

Measuring Linear Displacement with an LVDT

Linear displacement is movement in one direction along a single axis. The linear variable differential transformer (LVDT) is a sensor that measures linear displacement. LVDTs operate on the principle of a transformer and consist of a stationary coil assembly and moveable core. An LVDT measures displacement by associating a specific signal value for any given position of the core.

LVDT signal conditioners generate a sine wave for the primary output signal and synchronously demodulate the secondary output signal. The demodulated output is passed through a lowpass filter to remove high-frequency ripple. The resulting output is a DC voltage proportional to core displacement. The sign of the DC voltage indicates whether the displacement is to the left or right.

LVDTs typically come in 4-wire (open wire) and 5-wire (ratiometric wire) configurations. Wires from the sensor connect to a signal conditioning circuit that translates the output of the LVDT to a measurable voltage. The method of signal conditioning used differentiates the 4-wire and 5-wire configurations.

Measuring Resistance

Resistance is the opposition to passage of an electric current. One ohm is the resistance through which 1 V of electric force causes one ampere to flow. Two common methods for <u>measuring resistance</u> are the <u>2-wire method</u> and the <u>4-wire method</u>. For both methods, a current is sent through a resistor with a measurement device measuring the voltage drop from the signal across the resistor. The 2-wire method is easier to implement, but is not as accurate for resistances below 100 Ω . To calculate resistance, use the following formula:

$$R = V/I$$

where R = resistance, V = voltage, and I = current.

Measuring Sound Pressure

Sound pressure is the dynamic variation of the static pressure of air and is measured in force per unit area (Pa). The instantaneous sound pressure is typically averaged over a certain duration to give sound pressure level. Sound pressure level is normally represented on a logarithmic amplitude scale, which is similar to the human perception of hearing. Typical values on this logarithmic scale are a sound level of 0 dB, which is the average threshold of human hearing, 60 to 70 dB for normal conversation, 110 dB at an extremely loud concert, and 150 dB for the noise of a rocket takeoff or a jet engine at close range.

The Sound Pressure Level (SPL or LP) in decibels is defined as the following:

 $SPL = 20 \log 10 (p/pref)$

where p is the instantaneous sound pressure in Pa and pref is 20 μ Pa, the internationally accepted reference for sound pressure measurements, which roughly corresponds to the threshold of human hearing.

You can use a microphone to <u>measure sound pressure</u>. The microphone acts as a transducer, creating a voltage signal that is proportional to the instantaneous sound pressure.

Some analog input devices support programmable AC or DC coupling. If your device supports this feature, your sound pressure measurement task defaults to AC coupling. If you want a different coupling behavior, you can change this property programmatically.

Measuring Strain with a Strain Gage

Strain is the amount of deformation of a body due to an applied force. Specifically, strain is defined as the fractional change in length.

Strain can be positive (tensile) or negative (compressive). In practice, the magnitude of measured strain is very small. Therefore, strain is often expressed as microstrain ($\mu\epsilon$).

When a bar is strained with a uniaxial force, Poisson Strain causes the girth of the bar to contract in the transverse, or perpendicular, direction. The magnitude of this transverse contraction is a material property indicated by its Poisson's Ratio.

To <u>measure strain</u>, you typically use one or more strain gages in a Wheatstone bridge in one of several bridge configurations.

Using the Strain Gage Calibration Wizard

When you configure a strain measurement task, you can use the Strain Gage Calibration Wizard to calibrate your strain gage. Complete the following steps to calibrate the strain gage:

- 1. Click Calibrate to launch the Strain Gage Calibration Wizard.
- 2. Follow the steps on the Setup Hardware window. You can configure the following settings:
 - Enable Offset Nulling
 - Enable Shunt Calibration
- Click Next.
- 4. Use the Measure and Calibrate window to measure and calibrate your strain gage. The software automatically takes the first measurement, and the table displays the results. The table shows the following information:
 - Channel Information
 - Offset Adjustment—This section shows information about the offset error to help you determine if you should perform an offset null calibration. Perform an offset null calibration if you perform a shunt calibration.
 - ▶ Gain Adjustment—This section shows values related to gain error and the shunt calibration procedure:
- 5. Perform your measurement and calibration with the following buttons:
 - Click **Measure** to make a preliminary measurement using the offset and shunt calibration data from a previously run strain gage procedure.
 - Click **Reset Data** to reset the offset and shunt calibration data and make a measurement using their default values.
 - Click **Calibrate** to perform the offset and/or shunt calibration. The table shows the resulting measurements.
- 6. Click **Finish** when you are done. The calibration data is saved as part of your virtual channel. On successful completion, the software automatically uses the calibration data when scaling

voltage to strains on this virtual channel.



Note If you are using an SCXI-1520 with an SCXI-1314 terminal block, PXI-4220, SCXI-1121 with an SCXI-1321 terminal block, or an SCXI-1122 with an SCXI-1322 terminal block, refer to Device-Specific Strain Gage Information for more information about device-specific settings.

Related Topics

Overview of Strain Gages
Signal Conditioning Requirements for Strain Gages
Troubleshooting Your Strain Gage Calibration
High Accuracy Considerations

Device-Specific Strain Gage Calibration Information

The following devices have additional issues related to strain gage calibration.

SCXI-1520 (with SCXI-1314 terminal block), PXI-4220

Offset Nulling—These devices use hardware nulling circuitry. The resulting calibration values are stored in the AI.Bridge.Balance.CoarsePot and AI.Bridge.Balance.FinePot attributes of your strain virtual channel. Any residual offset value is stored in the AI.Bridge.InitialVoltage attribute of your strain virtual channel for additional software compensation.

SCXI-1121 (with SCXI-1321 terminal block)

Offset Nulling—Use the offset nulling potentiometer circuit on the SCXI-1321 to reduce initial offset before performing the calibration. Click **Measure** as you adjust the potentiometer to read the new offset value. Any residual offset value is stored in the AI.Bridge.InitialVoltage attribute of your strain virtual channel for additional software nulling compensation.

SCXI-1122 (with SCXI-1322 terminal block)

Offset Nulling—The hardware cannot perform offset nulling, so the offset value is stored in the AI.Bridge.InitialVoltage attribute of your strain virtual channel for software nulling compensation.

Troubleshooting Your Strain Gage Calibration

Refer to the following list if you receive an error while calibrating your strain gage:

- Make sure you have an actual strain gage connected to your physical channel, which is at rest (no strain).
- Check your strain attributes and make sure they reflect the capability of your measurement hardware and your strain gage setup.
- Make sure you have correctly connected your strain gage. Half-bridge and quarter-bridge setups require completion networks. If your measurement hardware has a built-in completion network, make sure you program it in software or set the completion jumpers in hardware. Additionally, quarter-bridge strain gage setup requires you to connect an external quarter-bridge completion resistor. Always use a precision completion-resistor whose value is equal to the nominal gage resistance of your quarter-bridge strain gage. Refer to your device documentation for details.
- Make sure you have specified the correct location of your shunt resistor. On some devices, the shunt resistor location is fixed. On other devices, you can choose the shunt resistor location. Refer to your device documentation to see where the shunt resistor is located by default.
- If you are drawing a lot of current through one or more strain gages, connect the remote sense inputs to your hardware (if supported).
- If you have long lead wires with large lead resistance, connect the remote sense to compensate for the voltage drop across the leads.
- Make sure your strain gage is properly bonded to the strain specimen.
- Make sure your strain gage does not have any short or open circuits.

High Accuracy Considerations

Your strain gage measurements can be inaccurate due to one or more factors. The following table lists these errors and ways to minimize each type of error.

Error Type	How to Minimize the Error
Gain	Perform frequent shunt calibration. Use a remote sensor to account for excitation voltage drop across lead wires. Use a strain gage with a higher precision gage factor. Accurately specify shunt resistance by manually measuring it.
Offset	Perform frequent offset nulling. Use precision completion resistors (for quarter-bridge and half-bridge setup). Check the gage resistance of your strain gage. Use a remote sensor to account for voltage drop across lead wires. Check the accuracy limitation of the offset nulling circuit.
Noise	Over-sample your data. Use a hardware lowpass filter. Use a software lowpass filter.
Temperature Drift	Perform frequent calibration. Use a temperature-compensated bridge type (all except quarter-bridge I).

Measuring Temperature with a Current Excitation Thermistor

A <u>thermistor</u> is a semiconductor made from metal oxides, pressed into a small bead, disk, or other shape at high temperatures, and coated with epoxy or glass.

As with an RTD, by passing a current through a thermistor, you can read the voltage across the thermistor and thus determine its temperature. Unlike RTDs, thermistors have a higher resistance (anywhere from 2,000 to 10,000 Ω) and a much higher sensitivity (~200 Ω /°C). However, thermistors are generally used to measure temperature only up to the 300° C temperature range.

Because thermistors have high resistance, lead-wire resistance does not affect the accuracy of the measurements. Unlike RTDs, <u>2-wire</u> measurements are adequate.

NI-DAQmx scales the resistance of a thermistor to a temperature using the Steinhart-Hart thermistor equation:

$$\frac{1}{T} = A + B(In(R)) + C(In(R))^3$$

where T is the temperature in Kelvins, R is the measured resistance, and A, B, and C are constants provided by the thermistor manufacturer.

Measuring Temperature with a Thermocouple

A <u>thermocouple</u> is created when two dissimilar metals touch, and the contact point produces a small open-circuit voltage that corresponds to temperature. Thermocouple measurements require sensing of the <u>cold-junction</u> temperature where the thermocouple wire is connected to the measurement system. Therefore, signal connection accessories should include an accurate cold-junction sensor, and should be designed to minimize any temperature gradients between the cold-junction sensor and thermocouple wire connections.

Other signal conditioning requirements include amplification and filtering, because thermocouples generate low-voltage signals and are susceptible to noise. Your system must also perform linearization through hardware or software because the output voltage is not linear with temperature.

Measuring Temperature with a Voltage Excitation Thermistor

A <u>thermistor</u> is a semiconductor made from metal oxides, pressed into a small bead, disk, or other shape at high temperatures, and coated with epoxy or glass.

As with an RTD, by passing a voltage through a thermistor, you can read the voltage across the thermistor and thus determine its temperature. Unlike RTDs, thermistors have a higher resistance (anywhere from 2,000 to 10,000 Ω) and a much higher sensitivity (~200 Ω /°C). However, thermistors are generally used to measure temperature only up to the 300° C temperature range.

Because thermistors have high resistance, lead-wire resistance does not affect the accuracy of the measurements. Unlike RTDs, <u>2-wire</u> measurements are adequate.

NI-DAQmx scales the resistance of a thermistor to a temperature using the Steinhart-Hart thermistor equation:

$$\frac{1}{T} = A + B(In(R)) + C(In(R))^3$$

where T is the temperature in Kelvins, R is the measured resistance, and A, B, and C are constants provided by the thermistor manufacturer.

Measuring Temperature with an RTD

An <u>RTD</u> is a temperature sensing device with resistance that increases with temperature.

Signal conditioning is generally required to measure temperature using an RTD. Because an RTD is a passive resistive device, you must pass a current through the device to produce a measurable voltage. Providing current to take a resistive measurement is a form of signal conditioning called current excitation. In addition to producing current excitation for the RTD, signal conditioning amplifies the output voltage signal, and filters the signal to remove unwanted noise. You also can use signal conditioning to electrically isolate the RTD and the monitored system from the DAQ system and the host computer.

Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common <u>voltage measurements</u> are DC and AC.

DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

Digital

You can use the digital ports and lines in your DAQ devices to acquire or generate digital samples. Refer to the following sections for more information:

Acquiring a Digital Sample (Line)

Acquiring a Digital Sample (Port)

Generating a Digital Sample (Line)

Generating a Digital Sample (Port)

Acquiring a Digital Sample (Line)

You can use the digital <u>lines</u> in your DAQ device to <u>acquire a digital value</u>. This acquisition is based on software timing. Each line corresponds to a virtual channel in your task.

Refer to the *NI-DAQmx Help* for information about <u>sample timing types</u> <u>and change detection</u>.

All E and M Series devices support TTL (transistor-transistor logic)-compatible digital signals. A TTL signal has the following two states:

- Logic Low = 0 V to +0.8 V
- Logic High = +2 V to +5 V

Common applications for immediate digital measurements include controlling relays and sensing external device states, such as the state of a switch.

Acquiring a Digital Sample (Port)

You can use the digital <u>port(s)</u> in your DAQ device to <u>acquire a digital</u> <u>value</u>. This acquisition is based on software timing. You can configure the ports individually to either measure or generate digital samples. Each port corresponds to a virtual channel in your task.

Refer to the *NI-DAQmx Help* for information about <u>sample timing types</u> and change detection.

All E and M Series devices support TTL (transistor-transistor logic)-compatible digital signals. A TTL signal has the following two states:

- Logic Low = 0 V to +0.8 V
- Logic High = +2 V to +5 V

Common applications for immediate digital measurements include controlling relays and sensing external device states, such as the state of a switch.

Generating a Digital Sample (Line)

You can use the digital <u>lines</u> in your DAQ device to <u>generate a digital sample</u>. This operation is based on software timing. Each line corresponds to a virtual channel in your task.

All E and M Series devices support TTL (transistor-transistor logic)-compatible digital signals. A TTL signal has two states, logic low and logic high:

```
Logic Low = 0 \text{ V} to +0.8 \text{ V}
Logic High = +2 \text{ V} to +5 \text{ V}
```

Common digital generation applications include controlling relays and driving external devices, such as an LED.

Generating a Digital Sample (Port)

You can use the digital <u>port(s)</u> in your DAQ device to <u>generate a digital sample</u>. This operation is based on software timing. On some devices, you can configure the ports individually to either measure or generate digital samples. Each port corresponds to a virtual channel in your task.

All E and M Series devices support TTL (transistor-transistor logic)-compatible digital signals. A TTL signal has the following two states:

- Logic Low = 0 V to +0.8 V
- Logic High = +2 V to +5 V

Common digital generation applications include controlling relays and driving external devices, such as an LED.

Counting Edges

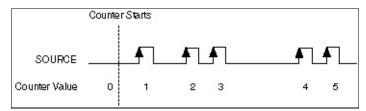
In <u>edge counting</u>, your device counts rising or falling edges using a counter. You can choose to do either single-point (on-demand or hardware timed) edge counting or buffered (sample clock) edge counting.

When doing on-demand edge counting, each subsequent read returns the number of edges counted since the counter was started. If you perform multiple reads without first starting the counter, the counter implicitly starts and stops with each Read function/VI call, and the number of counted edges is not cumulative between read calls.

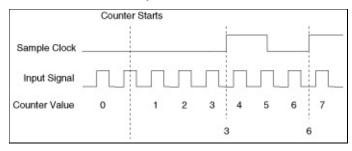
When doing hardware-timed edge counting, the read returns a value on each active edge of the sample clock. In this mode, you cannot specify a buffer. You must supply an external sample clock for hardware-timed edge counting.

When doing buffered edge counting, the number of edges counted is latched on each active edge of the sample clock and stored in the buffer. You must supply an external sample clock for buffered edge counting.

The following figure shows an example of edge counting in which the counter in a device counts five edges on the input terminal.



When doing buffered edge counting, the number of edges counted is latched on each active edge of the sample clock and stored in the buffer. There is no built-in clock for buffered edge counting, so you must supply an external sample clock.



Generating a Pulse

Using a counter to output pulses is called pulse generation. Some measurement devices can generate TTL pulses from the counter/timer of the device. The pulse is either low (starts high, pulses low, and returns high) or high (starts low, pulses high, and returns low). A pulse can be used as a clock signal, a gate, or a trigger for other measurements and generations.

Each pulse or pulse train consists of three parts:

- Initial Delay—The amount of time the output remains at the idle state before generating the pulse
- **High Time**—The amount of time the pulse is at a high level (5 V)
- Low Time—The amount of time the pulse is at a low level (0 V)

Using the DAQ Assistant, you can configure how long a pulse generated within a task remains at high level or low level.

The polarity of the pulse generation is controlled by **Idle State**. With **Idle State** set to **Low**, the pulse generation starts low for the **Initial Delay**, transitions high for the high time, and then transitions low for the low time. The high and low times are then repeated for the desired number of pulses. With **Idle State** set to **High**, the pulse generation starts high for the **Initial Delay**, transitions low for the low time, and then transitions high for the high time. The low and high times are then repeated for the desired number of pulses. In both cases, the output rests at **Idle State** after the generation completes.

Click here for more information about generating pulses.

Measuring Digital Frequency

You can use the counters of your device to <u>measure frequency</u>. To measure the frequency of a signal using the counters of your device, use the time base rate of your counter as a reference frequency. To configure your measurement, specify the expected range of the input signal, so that NI-DAQmx can choose its timebase to provide the highest resolution for the measurement. You can calculate the frequency of your signal based on the timebase rate and the number of counts using the following formula:

Frequency (in Hz) = Counter Timebase Rate / Count

To get the period of the signal, take inverse of the frequency. If, for example, the timebase rate of your measurement device is 100 MHz and you are counting 500 counts, the frequency of the input signal is 200kHz.

The Counter Timebase Rate is a known frequency (20 MHz or 100 kHz) and allows you to make frequency and time measurements. To configure a measurement, you specify the expected range of the input signal. Based on this range, NI-DAQmx automatically picks the internal timebase that provides the highest resolution for your measurement and uses it as the counter timebase.

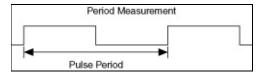
Depending on the phase of the input signal in relation to the start of the measurement, the first sample of continuous measurements is often invalid. For instance, if you are performing a continuous period measurement, and you start the measurement when the input signal is halfway through its current cycle, the measured period for the first sample is half of its expected value. Subsequent samples indicate the correct values because they are guaranteed to have seen a full period of the input signal. For this reason, the first sample of continuous period, pulsewidth, and semiperiod measurements often indicates a smaller value than the actual value. For continuous frequency measurements, the first sample often indicates a higher frequency than the actual frequency.

For frequency measurements, you also can perform measurements with one or two counters. For most applications, setting **Measurement**Method to 1 Counter (Low Frequency) is sufficient and desirable because it uses fewer resources. However, if you have a high-frequency or widely varying signal, you may wish to use one of the two counter

<u>measurement methods</u>—2 Counters (High Frequency) or 2 Counters (Large Range). Depending on the rate of your input signal and measurement method used, your measurement is subject to different quantization error. NI-DAQmx automatically internally routes as needed to perform the measurement across paired counters.

Measuring Period

You can use the counters on your measurement device to <u>measure</u> <u>period</u>. Period measurements measure the time between consecutive rising or falling edges.



The formula for period is as follows:

Period (in seconds) = Count / Counter Timebase Rate

where *Count* is the number of ticks of the counter timebase that elapsed during one period of the input signal being measured.

The Counter Timebase Rate is a known frequency (20 MHz or 100 kHz) and allows you to make frequency and time measurements. To configure a measurement, specify the expected range of the input signal. Based on this range, NI-DAQmx automatically chooses the internal timebase that provides the highest resolution for your measurement and uses it as the counter timebase.

Depending on the phase of the input signal in relation to the start of the measurement, the first sample of continuous measurements is often invalid. For instance, if you are performing a continuous period measurement, and you start the measurement when the input signal is halfway through its current cycle, the measured period for the first sample is half of its expected value. Subsequent samples indicate the correct values since they are guaranteed to have seen a full period of the input signal. For this reason, the first sample of continuous period, pulse-width, and semiperiod measurements often indicates a smaller value than the actual value. For continuous frequency measurements, the first sample often indicates a higher frequency than the actual frequency.

For period measurements, you also can perform measurements with one or two counters. For most applications, setting **Measurement Method** to **1 Counter (Low Frequency)** is sufficient and desirable because it uses fewer resources. However, if you have a high-frequency or widely varying signal, you may wish to use one of the <u>two counter measurement</u> <u>methods</u>—**2 Counters (High Frequency)** or **2 Counters (Large Range)**.

Depending on the rate of your input signal and measurement method used, your measurement is subject to different quantization error. NI-DAQmx automatically internally routes as needed to perform the measurement across paired counters.

Measuring Position with a Linear Encoder

On devices that use the NI-TIO timing chip, you can use the counters to <u>perform position measurements with encoders</u>. An encoder is a device that converts linear or rotary displacement into digital or pulse signals. Linear position can be measured with two-pulse encoders. You can choose to do either single-point (on-demand) position measurement or buffered (sample clock) position measurement.

The counter measures the position of the encoder using the A and B signals, which are offset by 90°. The NI-TIO counters also support the Z index, which provides a precise reference point and is available on some quadrature encoders.

To configure a measurement, you specify the initial sensor position through the Initial Position attribute/property, depending on your encoder type. You can also specify if the Z Input Terminal is used with the Z Index Enable attribute/property. The reload position upon a Z index signal, as well as when a Z index position should cause a reload to occur in relation to the A and B signals are also configurable through the Z Index Phase and Z Index Value attributes/properties, respectively.

When performing an on-demand position measurement, you first call the Start VI/function to arm the counter. Each subsequent read returns the current position of the encoder. If you perform multiple reads without first starting the counter, the counter implicitly starts and stops with each Read function/VI call, and the position is not recorded properly between read calls.

When doing buffered position measurement, the current position is latched on each active edge of the sample clock and stored in the buffer. There is no onboard clock for buffered position measurement, so you must supply an external sample clock.

Measuring Position with an Angular Encoder

On devices that use the NI-TIO timing chip, you can use the counters to <u>perform position measurements with encoders</u>. An encoder is a device that converts linear or rotary displacement into digital or pulse signals. Angular position can be measured with X1, X2, and X4 quadrature encoders. You can choose to do either single-point (on-demand) position measurement or buffered (sample clock) position measurement.

The counter measures the position of the encoder using the A and B signals, which are offset by 90°. The NI-TIO counters also support the Z index, which provides a precise reference point and is available on some quadrature encoders.

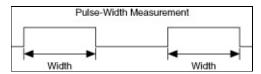
To configure a measurement, you specify the initial sensor position through the Initial Angle attribute/property, depending on your encoder type. You also can specify if the Z Input Terminal is used with the Z Index Enable attribute/property. The reload position upon a Z index signal, as well as when a Z index position should cause a reload to occur in relation to the A and B signals are also configurable through the Z Index Phase and Z Index Value attributes/properties, respectively.

When performing an on-demand position measurement, you first call the Start VI/function to arm the counter. Each subsequent read returns the current position of the encoder. If you perform multiple reads without first starting the counter, the counter implicitly starts and stops with each Read function/VI call, and the position is not recorded properly between read calls.

When doing buffered position measurement, the current position is latched on each active edge of the sample clock and stored in the buffer. There is no onboard clock for buffered position measurement, so you must supply an external sample clock.

Measuring Pulse Width

You can use the counters on your measurement device to measure pulse width. <u>Pulse-width measurements</u> measure the time between either a rising and a falling edge, or a falling and a rising edge.



The formula for pulse width is as follows:

Pulse width (in seconds) = Count / Counter Timebase Rate

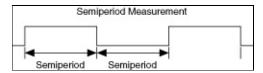
where *Count* is the number of ticks of the counter timebase that elapsed during one pulse width of the input signal being measured.

The Counter Timebase Rate is a known frequency (20 MHz or 100 kHz) and allows you to make frequency and time measurements. To configure a measurement, you specify the expected range of the input signal. Based on this range, NI-DAQmx automatically picks the internal timebase that provides the highest resolution for your measurement and uses it as the counter timebase.

Depending on the phase of the input signal in relation to the start of the measurement, the first sample of continuous measurements is often invalid. For instance, if you are performing a continuous period measurement, and you start the measurement when the input signal is halfway through its current cycle, the measured period for the first sample is half of its expected value. Subsequent samples indicate the correct values since they are guaranteed to have seen a full period of the input signal. For this reason, the first sample of continuous period, pulse-width, and semiperiod measurements often indicates a smaller value than the actual value. For continuous frequency measurements, the first sample often indicates a higher frequency than the actual frequency.

Measuring Semiperiod

You can use the counters on your measurement device to measure semiperiod. <u>Semiperiod measurements</u> measure the time between consecutive edges, alternating between high time and low time.



The formula for semiperiod is as follows:

Semiperiod(in seconds) = Count / Counter Timebase Rate (in Hz)

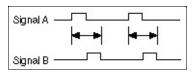
where *Count* is the number of ticks of the counter timebase that elapsed during the high or low time of the signal to measure.

The Counter Timebase is a clock with a known frequency (20 MHz or 100 kHz) and allows you to make frequency and time measurements. To configure a measurement, specify the expected range of the input signal. Based on this range, NI-DAQmx automatically chooses the internal timebase that provides the highest resolution for your measurement and uses it as the counter timebase.

Depending on the phase of the input signal in relation to the start of the measurement, the first sample of continuous measurements is often invalid. For instance, if you are performing a continuous period measurement, and you start the measurement when the input signal is halfway through its current cycle, the measured period for the first sample is half of its expected value. Subsequent samples indicate the correct values since they are guaranteed to have seen a full period of the input signal. For this reason, the first sample of continuous period, pulse-width, and semiperiod measurements often indicates a smaller value than the actual value. For continuous frequency measurements, the first sample often indicates a higher frequency than the actual frequency.

Measuring Two-Edge Separation

You can use the counters on your measurement device to measure twoedge separation. <u>Two-edge separation</u> is the time between the rising or falling edge of one digital signal (Signal A in the following figure) and the rising or falling edge of another digital signal (Signal B in figure).



The formula for two-edge separation is as follows:

Two-Edge Separation (in seconds) = Count / Counter Timebase Rate (in hertz)

where *Count* is the number of counter timebase ticks between the specified edge of the first signal and the specified edge of the second signal.

The *Counter Timebase Rate* is a known frequency for making frequency and time measurements. To configure a measurement, specify the expected range of the input signal. Based on this range, NI-DAQmx automatically selects the internal timebase that provides the highest resolution for your measurement and uses that timebase as the counter timebase.

Measurements Using Two Counters

You can measure digital frequency or the period of a TTL signal using two counters. Two methods are the high-frequency measurement method and the large range measurement method.

High-Frequency Measurement Method

If you measure the digital frequency or the period of a TTL signal with higher frequency, you can use the high-frequency method. To perform measurements using this method, NI-DAQmx uses the paired counter to generate a pulse train with a period specified using the **measurement time** attribute/property. The **measurement time** is generally much larger than the period of the input signal being measured to reduce <u>quantization error</u>. However, the **measurement time** must be small enough to keep the counter from rolling over. The measurement counter counts the number of periods of the input signal that occur during the **measurement time**, averages the results, and returns the averaged value in the Read function/VI. The value returned is calculated as follows:

Period (in seconds) = Measurement Time / Number of Periods Counted Frequency (in Hertz) = Number of Periods Counted / Measurement Time



Tip As you increase time, the accuracy of the reading also increases. However, if you specify too large a time, the counter may roll over, leading to an inaccurate frequency or period measurement.

Large Range Measurement Method

If you measure the digital frequency or the period of a TTL signal, you can use this two-counter method to measure signals with large ranges. This method is useful when you have a widely varying signal to measure and would like increased accuracy throughout the entire range. Refer to the <u>quantization error</u> section for more information on increasing measurement accuracy with the large-range measurement method. You can also use this method to measure signal frequencies that are faster than your counter timebase rate as long as the input signal does not exceed the maximum input frequency supported by the counter.

To perform measurements using this method, NI-DAQmx uses the paired counter to divide the input signal by a value specified by **Divisor**. However, you need to be careful the **Divisor** you choose does not cause the counter to roll over. This Divisor has the effect of shifting the measurable frequency range upward. For example, if you are using a 24-bit counter and your Counter Timebase Rate is 100 kHz, the measurable frequency range is approximately 0.006 Hz to 50 kHz. However, with a **Divisor** of 4, the measurable frequency range becomes 0.024 Hz to 200 kHz. The divided input signal is then measured using the low frequency one counter measurement method. The measured period is then scaled using the **Divisor**, and returned according to the following formula:

Period = Measured Period / Divisor

Frequency = Divisor / Measured Period



Tip As you increase the divisor, the accuracy of the reading also increases. However, if you specify too large a divisor, the counter may roll over, leading to an inaccurate frequency or period measurement.

Glossary

Prefixes	Numbers/Symbols	Α	В	C	D	E	F	G	Н	
JLM	NOPRS1		J \		V					

Symbol	Prefix	Value
n	nano	10 -9
μ	micro	10 -6
m	milli	10 -3
k	kilo	10 3
М	mega	10 6

Symbol	Meaning
%	percent
+	positive of, or plus
_	negative of, or minus
Ω	ohm
О	degree

Α

acceleration A change in velocity with respect to time.

accelerometer A sensor that represents acceleration as a voltage.

ADC Analog-to-digital converter—an electronic device, often

an integrated circuit, that converts an analog signal to a

digital value.

address A character code that identifies a specific location (or

series of locations) in memory.

ADE Application development environment—some examples

include LabVIEW and LabWindows/CVI.

advanced A terminal not accessible from the I/O connector or a

terminal terminal not commonly used in measurement

applications.

Al Analog input—acquisition of data.

amplification A type of signal conditioning that improves accuracy in

the resulting digitized signal by increasing signal

amplitude relative to noise.

analog Data represented by continuously variable physical

quantities.

AO Analog output—generation of data.

angular Movement about an axis, such as the angular motion of

displacement the shaft of a motor.

angular A device whose output signal represents the rotation of

displacement the shaft, such as a rotary variable differential

sensor transformer (RVDT).

API Application programming interface—A library of

functions, classes or VIs, attributes, and properties for

creating applications for your device.

asynchronous 1. Hardware—a signal that occurs or is acted upon

at an arbitrary time, without synchronization to

another signal, such as a reference clock.

2. Software—a VI or function that begins an operation and returns prior to the completion or termination of the operation.

attenuation

The reduction of a voltage or acoustical pressure. Measured referenced to the original voltage.

bandwidth The range of frequencies present in a signal, or the range of frequencies to which a measuring device can respond.

base A memory address that serves as the starting address for address programmable registers. All other addresses are located by adding to the base address.

bipolar A signal range that includes both positive and negative values (for example, 5 V to +5 V).

BIOS Basic Input/Output System—BIOS functions are the fundamental level of any PC or compatible computer. BIOS functions embody the basic operations needed for successful use of the computer hardware resources.

bit The smallest unit of data used in a digital operation. Bits are binary, so they can be either a 1 or a 0.

buffer In software, temporary storage for acquired or to-begenerated samples.

bus The group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC buses are the ISA bus and PCI bus.

C

C Series

A family of devices or modules used for analog input, analog output, digital input/output, and counter/timer applications. C Series devices work with chassis based on the CompactDAQ, CompactRIO, and other architectures, and are components of the NI USB-9XXX devices.

cDAQ

The prefix of the product model name of a CompactDAQ device, such as NI cDAQ-9172.

CH

Channel.

channel

- Physical—a terminal or pin at which you can measure or generate an analog or digital signal. A single physical channel can include more than one terminal, as in the case of a differential analog input channel or a digital port of eight lines. The name used for a counter physical channel is an exception because that physical channel name is not the name of the terminal where the counter measures or generates the digital signal.
- 2. Virtual—a collection of property settings that can include a name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. You can define NI-DAQmx virtual channels outside a task (global) or inside a task (local). Configuring virtual channels is optional in Traditional NI-DAQ (Legacy) and earlier versions, but is integral to every measurement you take in NI-DAQmx. In Traditional NI-DAQ (Legacy), you configure virtual channels in MAX. In NI-DAQmx, you can configure virtual channels either in MAX or in a program, and you can configure channels as part of a task or separately.

3. Switch—a switch channel represents any connection point on a switch. It may be made up of one or more signal wires (commonly one, two, or four), depending on the switch topology. A virtual channel cannot be created with a switch channel. Switch channels may be used only in the NI-DAQmx Switch functions and VIs.

clock A periodic digital signal.

CMRR Common-mode rejection ratio—a measure of the ability

of an instrument to reject interference from a common-

mode signal, usually expressed in decibels (dB).

code width The smallest detectable change in an input voltage of a

DAQ device.

cold-junction A method of compensating for inaccuracies in

compensation thermocouple circuits.

tree

CompactDAQ An architecture or chassis for C Series devices.

configuration Refers to the left window in MAX, which contains items

such as Data Neighborhood and Devices and Interfaces.

counter/timer A circuit that counts digital edges. Counters and timers

usually have from 16 bits to 48 bits (sometimes more) counting capability. The total number of counts possible equals 2N, where N is the number of bits in the counter.

When the edges counted are produced by a clock,

elapsed time can be computed from the number of edges

counted if the clock frequency is known.

convert clock The clock on a multiplexed device that directly causes

ADC conversions.

custom scale A method of instructing NI-DAQmx to apply additional

scaling to your data. Refer to the Create Scale

function/VI in your reference help.

D

DAC Digital-to-analog converter—an electronic device, often an

integrated circuit, that converts a digital value into a

corresponding analog voltage or current.

Refer to data acquisition. DAQ

A graphical interface for configuring measurement tasks, DAO

virtual channels, and scales. Assistant

DAQ device A device that acquires or generates data and can contain

multiple channels and conversion devices. DAO devices include plug-in devices, PCMCIA cards, and DAQPad devices, which connect to a computer USB or 1394 (FireWire) port. SCXI modules are considered DAQ

devices.

Samples. data

data acquisition (DAQ)

- 1. Acquiring and measuring analog or digital electrical signals from sensors, transducers, and test probes or fixtures.
- 2. Generating analog or digital electrical signals.

dB Decibel—the unit for expressing a logarithmic measure of

the ratio of two signal levels: dB=20log10 V1/V2, for

signals in volts.

DC direct current

delay from sample

The amount of time to wait after receiving a sample clock edge before beginning the acquisition of a sample.

delay from start

The amount of time to wait after receiving a Start Trigger

before beginning the operation.

determinism Characteristic of a system that describes how consistently it can respond to external events or perform operations within a given time limit.

device

1. An instrument or controller you can access as a single entity that controls or monitors real-world I/O points. A device often is connected to a host computer through some type of communication network.

2. See also DAO device and measurement device.

digital A TTL signal. Refer to edge.

DIO digital input/output

DMA direct memory access—A method of transferring data

between a buffer and a device that is used most often for

high-speed operations.

driver Software unique to the device or type of device, and

includes the set of commands the device accepts.

drop-down A graphical box with a down arrow button that lets you listbox

select values or options from a list. To select a value or

option in the selection box, click the down arrow for a complete list of values or options, then use your arrow

keys or mouse to select a value or option from the list.

DSUB D-subminiature connector

DUT device under test—a device used for testing purposes.

Ε

E Series A standard architecture for instrumentation-class,

multichannel data acquisition devices.

edge A digital edge is a single rising or falling TTL transition. An

analog edge is defined by the slope, level, and hysteresis

settings.

event A digital signal produced from a device or circuit. For an

advanced discussion of events, refer to **Events**.

excitation Supplying a voltage or current source to energize a sensor or

circuit.

F

- fall time The time for a signal to transition from 90% to 10% of the maximum signal amplitude.
- filtering A type of signal conditioning that you can use to remove unwanted frequency components from the signal you are measuring.
- FIFO A type of memory that implements a First In First Out strategy in which samples are removed in the order they were written. FIFOs are typically used as intermediate buffers between an ADC or DAC and the memory buffer.

floating Signal sources with voltage signals that are not connected to signal an absolute reference or system ground.

sources

G

gain

The factor by which a signal is amplified, often expressed in decibels (dB). Gain as a function of frequency is commonly referred to as the magnitude of the frequency response function.

signal

sources

grounded Signal sources with voltage signals that are referenced to a system ground, such as the earth or a building ground. Grounded signal sources are also called referenced signal sources.

Н

hardware The physical components of a computer system, such as the

circuit boards, plug-in boards, chassis, enclosures,

peripherals, and cables.

hardware A means of controlling signal generation. A digital signal,

timing such as a clock on a DAQ device, controls the rate of

generation.

hardware A form of triggering in which the source of the trigger is an

triggering analog or digital signal. Refer to **Software Trigger**.

hex Hexadecimal—a base-16 numbering system.

hysteresis A window around a trigger level that is often used to reduce

false triggering due to noise or jitter in the signal.

Hz Hertz—cycles per second of a periodic signal.

I

IEEE Family of IEEE standards defining a variety of smart

P1451 transducer interfaces. All of the standards within this family

support the concept of a TEDS that provides self-

identification and plug and play operation to transducers.

IEEE An IEEE standard that defines the concept of plug and play

P1451.4 sensors with analog signals. This is accomplished with the

addition of a TEDS in memory, typically an EEPROM, embedded within the sensor and communicated through a

simple, low-cost serial connection.

instrument Refer to driver.

driver

internal A physical channel not accessible from an I/O connector.

channel Internal channels are often used for calibration and are

intended for advanced applications.

interrupt A method whereby a device notifies the computer of some

condition on the device that requires the computer's attention. When this condition is a request for data or a notification of available data, interrupts are used as a data

transfer mechanism.

interrupt The relative priority at which a device can interrupt.

level

I/O Input/Output—the transfer of data to/from a computer

system involving communications channels, operator interface devices, and/or data acquisition and control

interfaces.

IRQ Interrupt ReQuest.

ISA Industry Standard Architecture—Also refers to a common

PC expansion bus.

isolation A type of signal conditioning in which you isolate the

transducer signals from the computer. Isolation makes sure the measurements from the measurement device are not

affected by differences in ground potentials.

J

jitter The amount of time that the loop cycle time varies from the desired time.

L

LED light-emitting diode—a semiconductor light source.

line An individual signal in a digital port. The difference

between a bit and a line is that the bit refers to the actual data transferred, and the line refers to the hardware the bit is transferred on. However, the terms line and bit are fairly interchangeable. For example, an 8-bit port is the

same as a port with eight lines.

linear Movement in one direction along a single axis.

displacement

linear A device that measures linear displacement.

displacement sensor

linearization A type of signal conditioning in which software linearizes

the voltage levels from transducers, so the voltages can

be scaled to measure physical phenomena.

LSB least significant bit—often used to refer to the smallest

voltage change detectable by an A/D converter or the smallest voltage change that can be generated by a D/A

converter.

LVDT Linear-voltage differential transformer—A sensor used to

measure linear displacement. An LVDT consists of a passive transform with one primary and two secondary windings. The primary winding is excited by an audio frequency range AC voltage, whose imbalance between

the secondary windings, is proportional to the

displacement. The secondary windings are identical, but are normally connected with opposite polarity, so the

transducer at resting position will have zero output

voltage.

M

M Series A standard architecture for instrumentation-class,

multichannel data acquisition devices.

MAX Measurement & Automation Explorer—A centralized

configuration environment that allows you to configure all

of your National Instruments devices.

device

measurement DAQ devices such as the M Series multifunction I/O (MIO) devices, SCXI signal conditioning modules, and

switch modules.

memory buffer

Refer to buffer.

memory mapping A technique for reading and writing to a device directly from your program, which avoids the overhead of delegating the reads and writes to kernel-level software. Delegation to the kernel is safer, but slower. Memory

mapping is less safe because an entire 4 KB page of memory must be exposed to your program for this to

work, but it is faster.

microphone A transducer that converts acoustical waves into

electrical signals.

MIO multifunction I/O—Designates a category of data

acquisition devices that have multiple analog input channels, digital I/O channels, timing, and optionally,

analog output channels. An MIO product can be

considered a miniature mixed signal tester, due to its broad range of signal types and flexibility. It is also known as multifunction DAQ. An E Series device is an

example of an MIO device.

module A board assembly and its associated mechanical parts,

front panel, optional shields, and so on. A module

contains everything required to occupy one or more slots

in a mainframe, SCXI and PXI devices are modules.

multiplexed An SCXI operating mode in which analog input channels mode are multiplexed into one module output so that the

cabled DAQ device has access to the multiplexed output as well as the outputs on all other multiplexed modules in

the chassis through the SCXI bus. Also called serial

mode.

multiplexer A switching device with multiple terminals that

sequentially connects each of its terminals to a single terminal, typically at high speeds. Often used to measure

several signals with a single analog input channel.

multithreading Running tasks of an application for a short amount of

time to give the impression of multiple tasks running

simultaneously.

Ν

NI-DAQ

Driver software included with all NI measurement devices. NI-DAQ is an extensive library of VIs and functions you can call from an application development environment (ADE), such as LabVIEW, to program all the features of an NI measurement device, such as configuring, acquiring and generating data from, and sending data to the device.

NI-DAQ 7.x Includes two NI-DAQ drivers—NI-DAQmx and Traditional NI-DAQ (Legacy)—each with its own API, hardware configuration, and software configuration.

NI-DAQmx The latest NI-DAQ driver with new VIs, functions, and development tools for controlling measurement devices. The advantages of NI-DAQmx over earlier versions of NI-DAQ include the DAQ Assistant for configuring channels and measurement tasks for your device for use in LabVIEW, LabWindows/CVI, and Measurement Studio; increased performance such as faster single-point analog I/O; and a simpler API for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.

Simulated Device

NI-DAQmx A replica of a device created using the **NI-DAQmx** Simulated Device option in the Create New menu of MAX for the purpose of operating a function or program without hardware. An NI-DAQmx simulated device behaves similarly to a physical device. Its driver is loaded, and programs using it are fully verified.

nonlinearity A measure in percentage of full-scale range (FSR) of the worst-case deviation from the ideal transfer function—a straight line.

> This specification is included only for DAQ products, such as signal conditioning products, that do not have an ADC. Because a product with this specification can also be used with a DAQ product with an ADC, this nonlinearity specification must be added to the relative accuracy

specification of the DAQ product with the ADC.

NRSE

Nonreferenced single-ended mode—all measurements are made with respect to a common (NRSE) measurement system reference, but the voltage at this reference can vary with respect to the measurement system ground.

0

onboard onboard channels	Provided by the data acquisition device. Channels provided by the plug-in data acquisition device.
onboard clock	The default source for a particular clock. Usually, the device has dedicated a circuit for producing this signal and its only purpose is to act as the source for a certain clock.
onboard memory	Memory provided by a device for temporary storage of input or output data. Typically, onboard memory is a FIFO, which is distinct from computer memory.
operating system	Base-level software that controls a computer, runs programs, interacts with users, and communicates with installed hardware or peripheral devices. Also referred to as OS.

P

parallel mode A type of SCXI operating mode in which the module sends each of its input channels directly to a separate analog input channel of the device connected to the module.

pattern I/O

pattern input and output—a digital I/O operation on which a clock signal initiates a digital transfer. Because the clock signal is a constant frequency, you can generate and receive patterns at a constant rate.

PCI

peripheral component interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. PCI has achieved widespread acceptance as a standard for PCs and work stations, and it offers a theoretical maximum transfer rate of 132 Mbytes/s.

An expansion bus architecture that has found widespread acceptance as a de facto standard in notebook-size computers. PCMCIA originated as a specification for add-on memory cards written by the Personal Computer Memory Card International Association.

PFI

PCMCIA

programmable function interface—general purpose input terminals, fixed purpose output terminals. The name of the fixed output signal is often placed on the I/O connector next to the terminal as a hint.

physical channel

Refer to channel.

PID

proportional integral derivative—Combination of proportional, integral, and derivative control actions. Refers to a control method in which the controller output is proportional to the error, its time history, and the rate at which it is changing. The error is the difference between the observed and desired values of a variable that is under control action.

pin

Refer to terminal.

The negative ratio of the strain in the transverse direction Poisson's Ratio (perpendicular to the force) to the strain in the axial direction (parallel to the force). A collection of digital lines. Usually the lines are grouped port into either a 8-bit or 32-bit port. Most E Series devices have one 8-bit port. port width The number of lines in a port. For example, most E Series devices have one port with eight lines; therefore, the port width is eight. Refer to <u>linear displacement sensor</u>. position sensor If there is no Reference Trigger, posttrigger samples are posttrigger the data acquired after the task is started. If there is a samples Reference Trigger, this is the data acquired after the Reference Trigger. plug and Devices that do not require DIP switches or jumpers to play devices configure resources on the devices. Also called switchless devices. plug and A transducer with an associated TEDS—includes both play sensors Virtual TEDS and smart TEDS sensors. Data acquired before the occurrence of the Reference pretrigger samples Trigger. pretriggering The technique used on a measurement device to keep a circular buffer filled with samples, so that when the Reference Trigger conditions are met, the buffer includes samples leading up to the trigger condition as well as

samples acquired immediately after the trigger.

programmed A data transfer mechanism in which a buffer is not used I/O and instead, the computer reads and writes directly to the device.

The amount of time required for a signal to pass through a propagation delay circuit.

A form of counter signal generation by which a pulse is pulsed

output generated when a counter reaches a certain value.

PWM pulse-width modulation

PXI PCI eXtensions for Instrumentation—a rugged, open

system for modular instrumentation based on

CompactPCI, with special mechanical, electrical, and software features. The PXI standard was originally developed by National Instruments in 1997 and is now

managed by the PXI Systems Alliance.

PXI trigger bus

The timing bus that connects PXI DAQ devices directly, by means of connectors built into the backplane of the PXI chassis, for precise synchronization of functions. This bus is functionally equivalent to the RTSI bus for PCI DAQ

devices.

R

range The minimum and maximum analog signal levels that the

ADC can digitize.

raw Data that has not been changed in any way. For input, data

is returned exactly as received from the device. For output, data is written as is to the device. Refer to <u>unscaled</u> and

scaled.

real time A property of an event or system in which samples are

processed as they are acquired instead of being

accumulated and processed at a later time.

referenced Signal sources with voltage signals that are referenced to a signal system ground, such as the earth or a building ground. Also

source called grounded signal sources.

resolution The smallest amount of input signal change that a device or

sensor can detect. The term *discrimination* is also used for

resolution.

rise time The time for a signal to transition from 10% to 90% of the

maximum signal amplitude.

route A connection between a pair of terminals. Any time the

source or destination terminal of a signal is specified, a

route is created.

RSE Referenced single-ended mode—all measurements are

made with respect to a common reference measurement system or a ground. Also called a grounded measurement

system.

RTD Resistance temperature detector—a metallic probe that

measures temperature based on its coefficient of resistivity.

RTSI bus Real-time system integration bus—the NI timing bus that

connects DAQ devices directly, by means of connectors on top of the devices, for precise synchronization of functions. This bus is functionally equivalent to the PXI Trigger bus for

PXI DAQ devices.

RVDT rotary variable differential transformer—a sensor whose output signal represents the rotation of the shaft.

S

s seconds

S samples. Refer to <u>sample</u>.

S/s samples per second—used to express the rate at which a

measurement device samples an analog signal.

sample A single measurement from a single channel or, for output,

a single generation to a single channel.

sample clock

The clock that initiates an acquisition of one sample from each channel in the scan list. For example, with each sample clock pulse, M Series devices acquire a sample on each analog input channel in a task by multiplexing each channel through a single ADC. On simultaneous sampling devices, the sample clock initiates the simultaneous acquisition of one sample from each channel in the task through a dedicated, per-channel ADC. No multiplexing (and therefore no convert clock) is necessary for S Series

devices.

sample

Refer to sample rate.

clock rate

sample rate The number of samples per channel per second. For

example, a sample rate of 10 S/s means sampling each

channel 10 times per second.

scale Data that has been mathematically transformed into

engineering units. Other manipulations also can be done

such as reordering to match the channel order.

scanning Method of sequentially connecting channels.

SCC Signal conditioning component—low channel count analog

or digital I/O modules for conditioning DAQ systems.

SCXI Signal Conditioning eXtensions for Instrumentation—the

NI product line for conditioning low-level signals within an external chassis near sensors so that only high-level

signals are sent to measurement devices in the noisy PC

	environment. SCXI is an open standard available for all vendors.
sensor	A device that responds to a physical stimulus (heat, light, sound, pressure, motion, flow, and so on) and produces a corresponding electrical signal.
signal	A means of conveying information. An analog waveform, a clock, and a single digital (TTL) edge are all examples of signals.
signal conditioning	The manipulation of signals to prepare them for digitizing.
smart TEDS sensor	A transducer with a built-in self-identification EEPROM that provides the TEDS.
software timing	A means of controlling signal generation. The software, such as NI-DAQmx, and the operating system control the rate of generation.
software trigger	A VI or function that, when it executes, triggers an action such as starting an acquisition.
source impedance	A parameter of signal sources that reflects current-driving ability of voltage sources (lower is better) and the voltage-driving ability of current sources (higher is better).
static AO	Analog output operations that use software timing.
static digital I/O	Software-timed digital I/O operations that do not involve the use of control signals in data transfers. Also known as software-timed I/O or unstrobed I/O.
strain	The amount of deformation of a body due to an applied force.
strobed I/O	Any operation in which every data transfer is timed by hardware signals. In the case of sample clock timing, this hardware signal is a clock edge. In the case of handshaking I/O, hardware signals involve two or three handshaking lines.
STC	system timing controller

synchronous 1. Hardware—a signal that occurs or is acted upon in

- synchrony with another signal, such as a reference clock.
- 2. Software—a VI or function that begins an operation and returns only when the operation is complete.

Т

task In NI-DAQmx, a collection of one or more channels,

timing, and triggering and other properties that apply to

the task itself. Conceptually, a task represents a measurement or generation you want to perform.

task buffer Refer to buffer.

TCR temperature coefficient of resistance—the average

resistance change per one degree at temperatures

between 0 °C and 100 °C.

TFDS transducer electronic data sheet—standardized data

structure, defined by IEEE 1451.4, for describing

sensors, typically stored in nonvolatile memory within a sensor. The manufacturer of the sensor stores, into this memory, initial information such as manufacturer name,

sensor type, model number, serial number, and

calibration data. The TEDS data structure also includes

space for custom information such as channel ID,

location, position, direction, tag number, etc. Alternatively, the TEDS data may be stored in a file or database record

as a Virtual TEDS. For information on IEEE 1451.4compliant TEDS sensors, refer to www.ni.com/pnp.

Sensor

TEDS Class I A smart TEDS sensor with a constant-current powered transducer with a two-wire interface, such as an

> accelerometer. Class I transducers also include diodes or analog switches with which the multiplexing of the analog signal with the digital TEDS information on the single pair of wires is possible. The digital portion of the mixed-mode

interface (Class 1 or Class 2) is based on the 1-Wire

protocol from Maxim/Dallas Semiconductor.

TEDS Class II Sensor

A smart TEDS sensor with separate wires for the analog and digital portions of the TEDS mixed-mode interface.

The analog input/output of the transducer is left unmodified, and the digital TEDS circuit is added in

parallel, such as thermocouples, RTDs, and bridge-based

sensors. The digital portion of the mixed-mode interface (Class 1 or Class 2) is based on the 1-Wire protocol from Maxim/Dallas Semiconductor.

terminal A named location on a DAQ device where a signal is

either generated (output or produced) or acquired (input

or consumed).

count

terminal When counting up, an N bit counter reaches its terminal

count at 2N -1. An N bit counter counting down reaches

its terminal count at 0.

thermistor A semiconductor sensor that produces a repeatable

change in electrical resistance as a function of temperature. Most thermistors have a negative

temperature coefficient.

thermocouple A temperature sensor created by joining two dissimilar

metals. The junction produces a small voltage as a

function of the temperature.

threshold The voltage level a signal must reach for a trigger to

occur.

tick A digital edge of a clock.

timebase A clock that is divided down to produce another clock or a

clock provided to a counter for measuring elapsed time.

Traditional An upgrade of the earlier version of NI-DAQ. Traditional

NI-DAQ (Legacy) has the same VIs and functions and

(Legacy) works the same way as NI-DAQ 6.9.x, except you can

use both Traditional NI-DAQ (Legacy) and NI-DAQmx on the same computer, and some hardware is no longer

upported

supported.

transducer Refer to sensor.

transducer A type of signal conditioning that uses external voltages

excitation and currents to excite the circuitry of a signal conditioning

system into measuring physical phenomena.

trigger Any signal that causes a device to perform an action,

such as starting an acquisition.

TTL Transistor-transistor logic—a signal having two discrete levels, a high and a low level.

U

unipolar A signal range that is always positive (for example, 0 to +10 V).

unscaled Samples in the integer form that the hardware produces or requires. Although no mathematical transformations are applied to unscaled data, other manipulations may be done such as reordering to match the channel order.

unstrobed Refer to static digital I/O. I/O

USB A USB-based family of devices used for analog input, analog output, digital input/output, and counter/timer applications.

Some example devices include the NI USB-9201, NI USB-9211, NI USB-9215, NI USB-9221, NI USB-9233, and NI USB-9237 devices. These devices are also referred to as USB DAQ with Integrated Signal Conditioning.

V

V volts

VI Refer to <u>virtual instrument</u>.

virtual Refer to channel.

channel

virtual A program in LabVIEW that models the appearance and

instrument function of a physical instrument.

VISA Virtual Instrumentation Software Architecture.

W

waveform data A LabVIEW data type that bundles timing information type along with the data.

WDT Refer to <u>waveform data type</u>.

Important Information

Warranty

Copyright

Trademarks

Patents

Warning Regarding Use of NI Products

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

Except as specified herein, National Instruments makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. Customer's right to recover damages caused by fault or negligence on the part of National Instruments shall be limited to the amount theretofore paid by the customer. National Instruments will not be liable for damages resulting from loss of data, profits, use of products, or incidental or consequential damages, even if advised of the possibility thereof. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action

accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about <u>National</u> Instruments trademarks.

FireWire® is the registered trademark of Apple Computer, Inc.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix® and Tek are registered trademarks of Tektronix, Inc.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the patents.txt file on your CD, or <u>ni.com/patents</u>.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

- (1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.
- (2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS. INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR

DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES. INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- <u>Support</u>—Online technical support resources at ni.com/support include the following:
 - Self-Help Resources—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable <u>KnowledgeBase</u>, <u>product</u> <u>manuals</u>, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
 - Free Technical Support—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the NI Developer Exchange at ni.com/exchange. National Instruments Applications Engineers make sure every question receives an answer.

For information about other <u>technical support options</u> in your area, visit ni.com/services or <u>contact</u> your local office at ni.com/contact.

- <u>Training and Certification</u>—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- <u>System Integration</u>—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your <u>local office</u> or NI corporate headquarters. You also can visit the <u>Worldwide Offices</u> section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Branch Offices

Office	Telephone Number
Australia	1800 300 800
Austria	43 662 457990-0
Belgium	32 (0) 2 757 0020
Brazil	55 11 3262 3599
Canada	800 433 3488
China	86 21 5050 9800
Czech Republic	420 224 235 774
Denmark	45 45 76 26 00
Finland	358 (0) 9 725 72511
France	33 (0) 1 57 66 24 24
Germany	49 89 7413130
India	91 80 41190000
Israel	972 0 3 6393737
Italy	39 02 41309277
Japan	0120-527196 / 81 3 5472 2970
Korea	82 02 3451 3400
Lebanon	961 (0) 1 33 28 28
Malaysia	1800 887710
Mexico	01 800 010 0793
Netherlands	31 (0) 348 433 466
New Zealand	0800 553 322
Norway	47 (0) 66 90 76 60
Poland	48 22 3390150
Portugal	351 210 311 210

Russia	7 495 783 6851
Singapore	1800 226 5886
Slovenia	386 3 425 42 00
South Africa	27 0 11 805 8197
Spain	34 91 640 0085
Sweden	46 (0) 8 587 895 00
Switzerland	41 56 2005151
Taiwan	886 02 2377 2222
Thailand	662 278 6777
Turkey	90 212 279 3031
United Kingdom	44 (0) 1635 523545
United States (Corporate)	512 683 0100